RANS-Based Aerodynamic Shape Optimization of a Strut-Braced Wing with Overset Meshes

Ney R. Secco and Joaquim R. R. A. Martins

Abstract The strut-braced wing aircraft configuration promises to reduce fuel burn by enabling higher spans that reduce lift-induced drag. A successful design for this configuration depends on a careful trade-off between the various sources of drag and structural weight. When using CFD-based aerodynamic shape optimization, generating high-quality structured meshes for the strut-braced wing configuration becomes challenging, especially near junctions. Furthermore, mesh deformation procedures frequently generate negative volume cells when applied to these structured meshes during optimization. We address this issue by using overset meshes and a component-based parametrization technique to achieve a flexible design optimization cycle capable of handling changing junctions. We use this approach to minimize drag of the PADRI 2017 strut-braced wing benchmark for a fixed lift constraint at transonic flight conditions. The drag of the optimized configuration is 15% lower than the baseline due to the reduction of shocks and separation in the wing-strut junction region. This result represents an example in which high-fidelity modeling is required to quantify the benefits of a new aircraft configuration and address potential issues during the conceptual design.

Contents

1 Introduction

One approach to decrease aircraft fuel burn is the reduction of induced drag through longer wing spans. However, when aerostructural trade-offs are considered, there is a limit to the span that can be achieved before drag reduction is overwhelmed by structural weight increase [1, 2]. The strut-braced wing (SBW) configuration addresses this issue by adding a strut to alleviate the bending moment in the main wing. The additional strut allows the wing to have a longer span, lowering induced drag [3]. The strut also enables the reduction of the main wing thickness-to-chord ratio, thus decreasing structural weight and wave drag. Despite these benefits, the wing-strut junction increases interference drag; thus, the feasibility of this type of aircraft depends heavily on the synergy between the wing and the strut, as well as on the careful design of the junctions.

The truss-braced wing (TBW) is a variant of the SBW configuration in which the main wing has additional supporting elements besides the strut, called juries [4]. The juries are used to stabilize buckling modes of the wing and strut [5]. However, the additional junctions and the blockage effect of the juries in the wing-strut gap result in even more complex trade-offs between aerodynamics and structures.

The Hurel–Dubois HD-34, built in the 1950's and primarily used for aerial photography, was one of the first TBW airplanes to be successfully flown [6]. Since then, multiple studies have been conducted to verify the feasibility of this configuration for commercial transport, motivated by the desire to reduce fuel burn. For instance, a study conducted by NASA in 1980 concluded that the SBW configuration would have a lower fuel consumption when compared to conventional configurations, though the loss in productivity due to lower cruise speeds and increase in costs associated with the wing size and complexity could hinder the practicality of this configuration [7].

Despite these concerns, studies of SBWs and TBWs continued as new analysis and design tools became available. Efforts in the late 1990's and early 2000's assessed the feasibility of the SBW in a commercial transport mission profile using multidisciplinary tools involving aerodynamics, structures, propulsion, and stability considerations [8, 9]. The complexity of the multidisciplinary environment often forced the use of medium-fidelity and semi-empirical aerodynamics, such as discrete vortex method for induced drag [10], Korn equation for wave drag, and skin-friction coefficients for parasite drag to obtain a tractable optimization problem. In some efforts, the interference drag associated with junctions was estimated via surrogate models [11, 12] for conceptual design of SBW and TBW configurations [3]. These models were often nonideal as the computational cost associated with the surrogate model generation could limit the number of design parameters and their bounds.

More recently, NASA and Boeing have conducted analyses of the TBW configuration using both wind tunnel tests and computational fluid dynamics (CFD) simulations as part of the SUGAR (Subsonic Ultra-Green Aircraft Research) project [5]. One TBW configuration may even be included in the next-generation of NASA X-planes to verify its potential for fuel burn reduction [13].

This increasing interest and investment in these configurations justifies efforts to increase the fidelity of models used in SBW and TBW analyses to understand in more detail the aerodynamic and structural interaction among the aircraft components and to correctly quantify the potential performance gains. For instance, Carrier et al. [6] used a Reynolds-averaged Navier–Stokes (RANS) solver to compute aerodynamic loads on the main wing which were then coupled with a finite-element model to compute structural deformation. Gagnon and Zingg [14, 15] performed Euler-based aerodynamic shape optimization of a SBW considering both the main wing and strut.

In a previous study, we conducted a RANS-based aerodynamic shape optimization (ASO) of a TBW in the transonic regime by using shape and twist design variables [16]. Though this study offered insights into the performance of TBW configurations, it was limited in several respects that have since been addressed.

Firstly, it utilized a structured multiblock mesh for CFD analyses that prevented changes in the wing-strut junction geometry due to complications in deforming the mesh near the concave regions of the junctions without generating negative volume cells. We have since extended our CFD solver to handle overset meshes [17]. The overset methodology consists in having multiple overlapped meshes in space and then interpolating information among them to reconstruct the overall flow field. This approach allows the independent generation of high-quality meshes fitted to each component and junction using, for instance, hyperbolic mesh extrusion [18]. Since overset meshes consist of overlapping structured meshes, the advantages of this type of mesh, such as memory efficiency and mesh quality, still apply [19].

Overset meshes have already been used in ASO applications. Liao and Tsai [20] used overset meshes for Euler-based optimization of airfoils using the continuous adjoint approach. Lee and Kim [21] implemented the discrete adjoint method for an Euler-based CFD solver and optimized the wing of a wing-body aircraft configuration using overset meshes. Lee et al. [22] later extended their methodology to the RANS equations and the k-ω turbulence model to minimize distortion in a boundary-layer-ingestion engine inlet. Kenway et al. [17] used overset meshes and the hybrid adjoint method to optimize the wing of the Common Research Model (CRM) aircraft configuration while using RANS analysis with the Spalart–Allmaras (SA) turbulence model.

Secondly, our previous TBW optimization was limited by characteristics of our geometry manipulation module. The previous geometry manipulation method operated on the entire geometry, rather than on individual components. Therefore, it was not possible to implement independent sets of design variables for each component. For instance, if we changed the spanwise wing-strut intersection position, the lower surface of the wing would be moved as well, causing unwanted compression and extension of surface cells on the wing CFD mesh. We recently developed a component-based geometry manipulation module that allows the strut to freely move with respect to the wing, avoiding the prior geometry limitations [23].

The goal of this work is to perform RANS-based aerodynamic shape optimization of a SBW in the transonic regime using overset meshes and component-based geometry manipulation to allow wing-strut junction shape design. We use the baseline SBW geometry and flow conditions from the Platform for Aircraft Drag Reduction Innovation workshop (PADRI 2017) [24]. This geometry, shown in Fig. 1, consists of three primary components: fuselage, wing, and strut. The fuselage geometry and the wing-fuselage junction is kept unchanged in this work. Table 1 shows the main dimensions of each component and the flow condition used for aerodynamic analysis.

We use a RANS overset solver for aerodynamic analysis that allows us to capture viscous effects, such as separation around the junctions. The outer-mold line (OML) of each primary component is modified via free-form deformation (FFD) boxes, and the intersections among these components are recomputed after every geometry update using triangulated surfaces. The RANS analyses provide a more accurate quantification of the drag reductions that are possible for the SBW, thus improving upon Euler-based studies [14, 15, 25] and complementing previous low-fidelity multidisciplinary design optimization estimates [3, 4, 8, 9, 26].

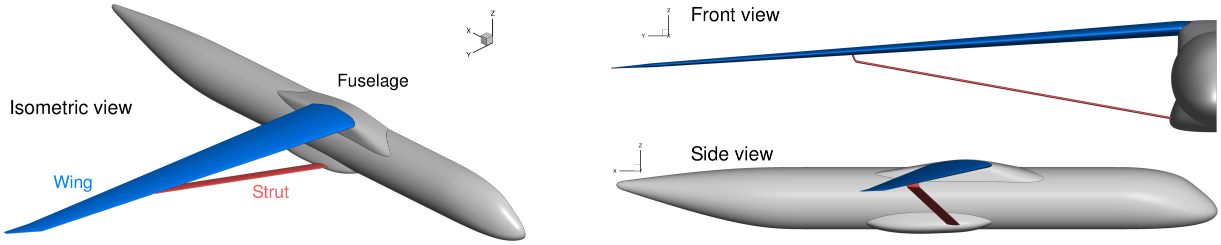


Figure 1: Baseline SBW configuration of the PADRI 2017 workshop. Views are in different scales.

2 Optimization Framework

High-fidelity aircraft optimization requires a multidisciplinary framework that should be computationally efficient and massively parallelized due to the high computational cost of each analysis. Gradient-based optimization is a must due to the large number of design variables [27]; thus, this framework should be capable of efficiently computing derivatives of the objective and constraint functions with respect to all design variables.

The MACH framework (multidisciplinary design optimization for aircraft configurations with high fidelity) meets these needs [28]. Even though MACH has support for aerostructural optimization, this work will exclusively utilize the modules required for aerodynamic shape optimization. These include the geometry

Table 1: Geometric characteristics and flight condition used for the baseline SBW configuration analysis.

Wing		**Strut**		**Fuselage**	
Planform area [m^2]	161	Planform area [m^2]	28.9	Length [m]	41.7
Span [m]	55.6	Span [m]	33.4	Diameter [m]	4.3
Mean aerodynamic chord [m]	3.264	Mean aerodynamic chord [m]	0.865		
Aspect ratio	19.2	Aspect ratio	38.6		
Taper ratio	0.256	Taper ratio	1.00		
1/4 Chord sweep [deg]	15.0	1/4 Chord sweep [deg]	10.9		
Dihedral [deg]	-4.0	Dihedral [deg]	10.2		

Flight Condition	
Mach	0.72
α [deg]	1.0
Altitude [ft]	30,000

manipulator (pyGeo), the collar mesh generator (pySurf), the volume mesh deformation module (pyWarp), and the flow solver (ADflow). We use these modules in each design iteration cycle to compute function values and their gradients, which are then provided to the numerical optimizer (SNOPT). We also use a volume mesh generation module (pyHyp) to generate the initial CFD meshes used throughout the optimization.

The optimization framework requires three groups of inputs for each primary component of the geometry:

Structured surface mesh (Fig. 2), which is used to generate the volume mesh by the hyperbolic extrusion module (pyHyp). The mesh should be constructed with the desired resolution for the CFD analysis and refined near intersections to facilitate the overset hole cutting process.

Triangulated (unstructured) surface mesh (Fig. 3), which is used as the reference for intersection computation and collar mesh generation by the collar mesh generation module (pySurf). This triangulated mesh should be finer than the structured surface mesh in order to appropriately represent high-curvature regions (such as leading edges) for the collar mesh generation process.

Free-form deformation (FFD) box (Fig. 4), which envelops the surfaces described above and whose control points give the necessary shape control resolution for the design optimization.

The overall computational framework arrangement is shown in the extended design structure matrix (XDSM) [29] diagram of Fig. 5. Elements with the (0) superscript refer to the baseline configuration, while the $*$ superscript denotes optimized values. Stacked elements indicate that the given data or module should be defined for each aircraft component and collar mesh. This figure also highlights how the volume mesh generation process is only used at the initialization step and then replaced by the mesh deformation module during the optimization. The framework modules are described in the next sections, including utilization details of the three sets of inputs. Further details on the MACH framework are provided by Kenway et al. [28].

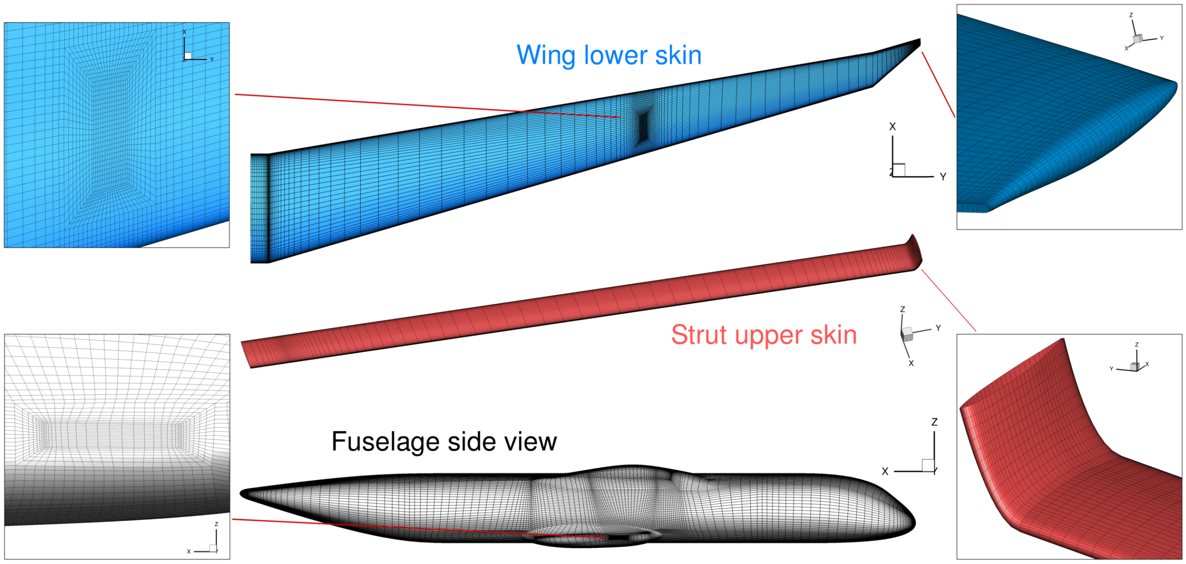


Figure 2: Structured surfaces meshes of the primary components of the aircraft. The O-grids near intersections increase the cell density to facilitate the overset hole cutting process.

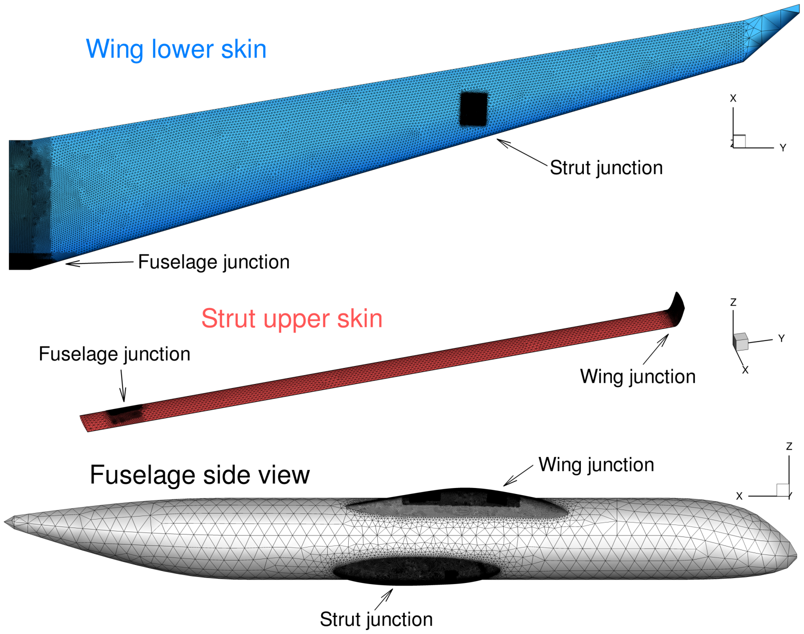


Figure 3: Triangulated surfaces used for intersection detection and collar mesh generation.

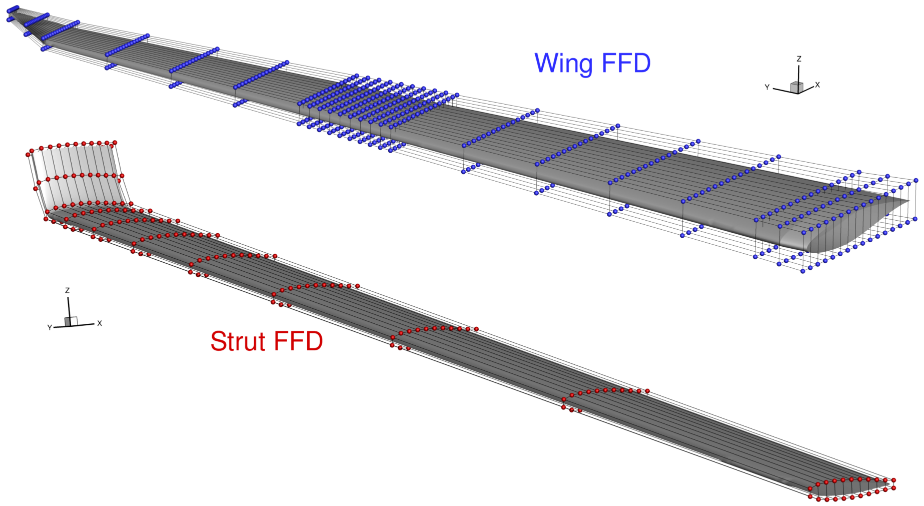


Figure 4: FFD boxes of the primary components whose shape will be optimized. The dots represent the FFD box control points.

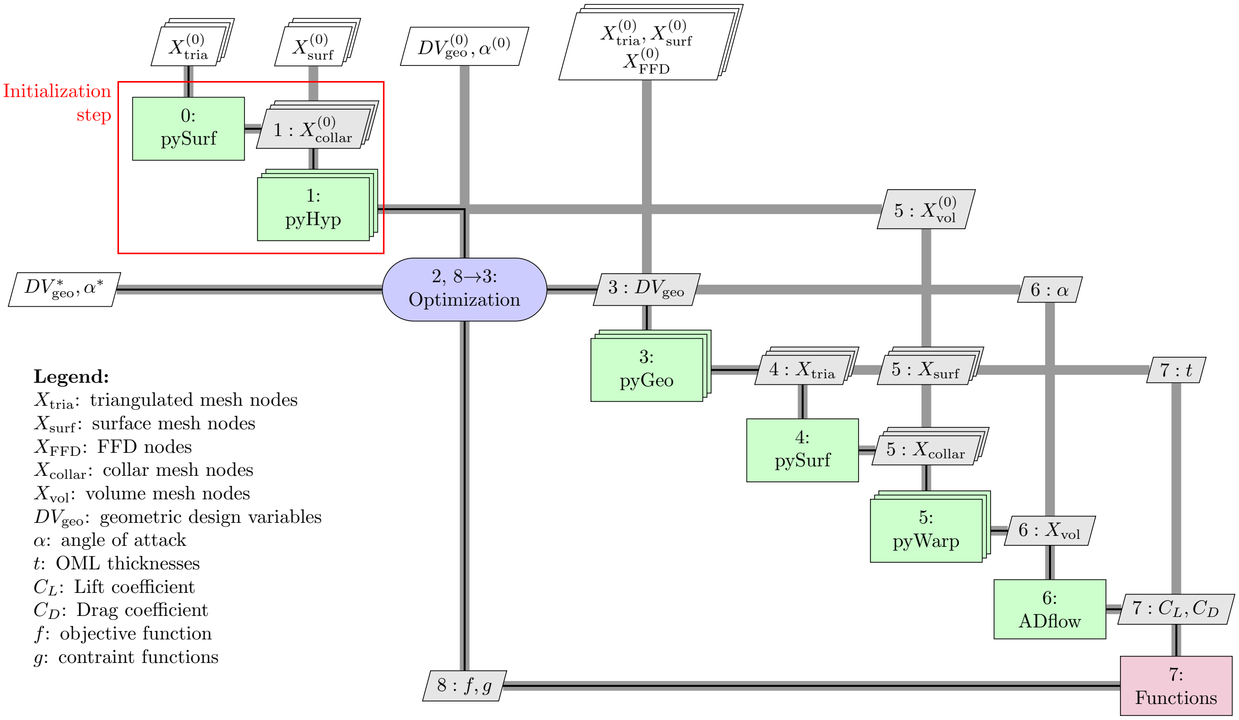

$X^{(0)}_{\text{tria}}$
$X^{(0)}_{\text{surf}}$
$DV^{(0)}_{\text{geo}}, \alpha^{(0)}$
$X^{(0)}_{\text{tria}}, X^{(0)}_{\text{surf}}$ $X^{(0)}_{\text{FFD}}$
Initialization step
0: pySurf
$1 : X^{(0)}_{\text{collar}}$
1: pyHyp
$5 : X^{(0)}_{\text{vol}}$
$DV^{*}_{\text{geo}}, \alpha^{*}$
2, 8→3: Optimization
$3 : DV_{\text{geo}}$
$6 : \alpha$
3: pyGeo
$4 : X_{\text{tria}}$
$5 : X_{\text{surf}}$
$7 : t$
4: pySurf
$5 : X_{\text{collar}}$
5: pyWarp
$6 : X_{\text{vol}}$
6: ADflow
$7 : C_L, C_D$
$8 : f, g$
7: Functions
Legend:
X_{tria}: triangulated mesh nodes
X_{surf}: surface mesh nodes
X_{FFD}: FFD nodes
X_{collar}: collar mesh nodes
X_{vol}: volume mesh nodes
DV_{geo}: geometric design variables
α: angle of attack
t: OML thicknesses
C_L: Lift coefficient
C_D: Drag coefficient
f: objective function
g: contraint functions

Figure 5: XDSM of the overall optimization framework. The initialization step generates the reference volume nodes for the mesh deformation.

2.1 Geometry manipulation—pyGeo

pyGeo is a Python module capable of manipulating geometries with the FFD approach [30]. It uses structured hexahedral boxes of control points (which we call FFD boxes) surrounding the geometry that we want to manipulate to define a tri-variate B-spline mapping in the space. We then *embed* the structured surface meshes and the triangulated meshes of the geometry by determining the parametric coordinates of each surface node in these B-spline boxes. Any changes done to the control points can be transferred to the embedded nodes via the B-spline mapping. With our aim of optimizing the shapes of the wing and the strut of the SBW, we need to create two FFD boxes, one for each component (Fig. 4).

The positions of the FFD box control points become the primary shape variables of the problem. In addition, pyGeo can translate high-level design variables such as twist, sweep, and span by simultaneously moving several control points in a consistent manner. We describe the design variables defined for the SBW optimization in more detail in Sec. 3.

The FFD method does not generate a new geometry by itself: it parametrizes displacements of the embedded mesh nodes, allowing the optimization framework to be agnostic with respect to geometry generation tools. Furthermore, the FFD method is suitable for gradient-based optimization since the B-spline mapping is analytically differentiable [31].

We need to embed the surface nodes of the structured meshes and of the triangulated meshes into the same FFD box so that both surface representations are subject to consistent deformations during every optimization cycle. The updated triangulated surfaces are used by the pySurf module to regenerate surface collar meshes, while the updated structured surface meshes are used by the mesh deformation module (pyWarp) to update the volume nodes.

pyGeo can also measure the thicknesses of the embedded components. This allows us to define thickness constraints to prevent unfeasible designs from the structural point of view, as we discuss in Sec. 3.

2.2 Collar mesh generation—pySurf

The use of overset meshes requires that each intersection between primary components have its own dedicated mesh, which is called the *collar mesh* [32]. This specialized mesh is responsible for accurately representing the intersection geometry. Since pyGeo changes each primary component independently, we need to recompute intersections among these components to appropriately update the collar meshes, which is achieved by the pySurf module [23].

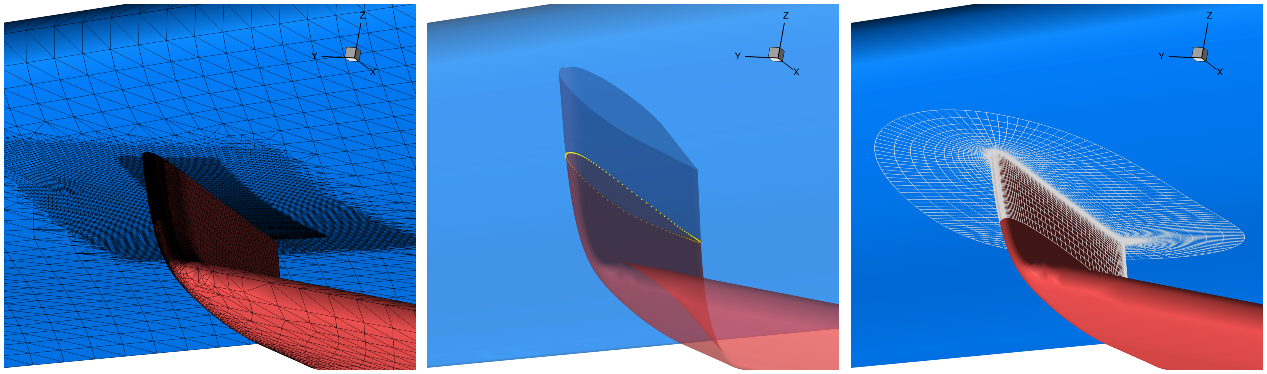
pySurf uses the triangulated surfaces representing each primary component to detect intersections. In every optimization iteration, pyGeo updates the triangulated surface nodes using FFD. Then pySurf uses the updated triangulated surfaces (Fig. 6a) to find intersection curves among the components (Fig. 6b). We use an alternating digital tree (ADT) search [33] to efficiently detect candidate intersecting triangles, upon which we apply the actual triangle-triangle intersection algorithm [34].

Once we have the new intersection curve, we use hyperbolic surface mesh marching algorithms [35] to grow structured surface meshes on each side of the intersection (Fig. 6c). We always distribute the same number of points along the intersection curve prior to the mesh marching to ensure that the mesh topology and the number of surface nodes remain the same throughout the optimization.

Before the optimization, the automatically generated surface collar mesh is used by the volume mesh generation module (pyHyp) to generate the baseline volume collar mesh, as shown in Fig. 5. In contrast, during the optimization, the mesh deformation module (pyWarp) uses the updated position of the collar mesh surface nodes to deform the cells of the volume mesh. This procedure avoids the mesh regeneration process, which is costly and generates discontinuities in the design space.

We apply automatic differentiation (AD) to the intersection detection and surface mesh marching algorithms to generate an equivalent source code that computes derivatives of the collar mesh nodes with respect to the triangulated surface nodes. Because the number of input variables of the overall optimization problem is greater than the number of functions of interest, we generate the differentiated code with the reverse mode of the Tapenade AD engine [36].

For the purposes of this work, we need to generate one triangulated surface for each primary component of the SBW (Fig. 3). The SBW arrangement results in three intersections among these components and thus requires the corresponding collar meshes, as shown in Fig. 7 for the case of the baseline geometry.



a) Triangulated surfaces b) Intersection curve c) Hyperbolic mesh marching

Figure 6: Steps of the surface collar mesh generation for the wing-strut intersection.

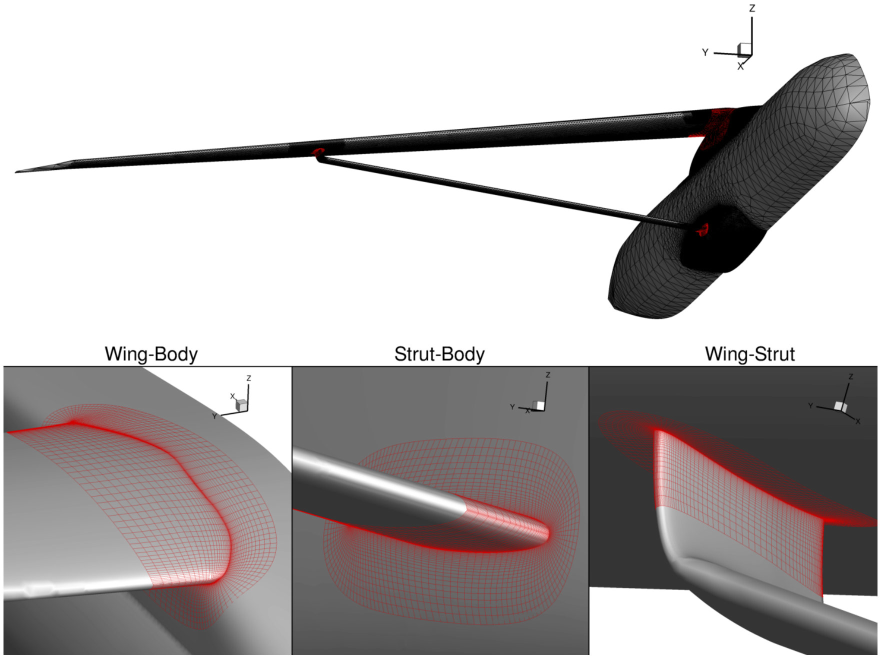
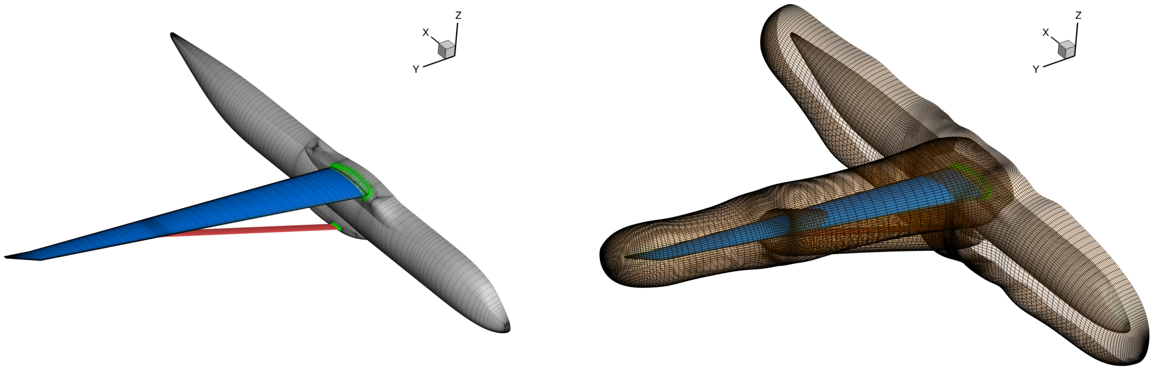


Figure 7: Collar meshes generated over the triangulated surfaces of the SBW by pySurf.

2.3 Volume mesh generation—pyHyp

The CFD solver used in this work requires a structured volume mesh fitted to the airplane OML. We use hyperbolic volume mesh marching schemes [18] to extrude structured surface meshes into volume meshes. We apply this methodology to the user-provided structured surface meshes of each primary component (Fig. 2) and to the surface collar meshes generated by pySurf (Fig. 7). These surface meshes should meet the required resolution for CFD simulations, for instance, by refining leading edges, trailing edges, and junction areas.

The mesh extrusion for the SBW configuration is shown in Fig. 8. This step is completed at the beginning of the optimization to generate the baseline volume meshes, as indicated by the initialization step



a) Initial surface meshes

b) Extruded near-field meshes

Figure 8: Hyperbolic extrusion of the surface meshes into volume meshes.

of Fig. 5. These meshes are subsequently deformed by the mesh deformation module (pyWarp) throughout the optimization, as this process is computationally less expensive than regenerating new volume meshes at each design point. Therefore, there is no need to differentiate the pyHyp module.

The surface meshes are extruded to a small extension, from 1 to 3 mean aerodynamic chords; the overset interpolation boundary conditions are then applied at the external surface of those meshes. Then, pyHyp uses the bounding box of these near-field volume meshes to create a background mesh that reaches an appropriate distance for the far-field boundary conditions. The background mesh is generated in two steps. First, we compute the bounding box defined by the near-field meshes and fill this box with Cartesian cells based on an user-provided desired cell size. Next, we use hyperbolic extrusion to march the exposed surface of this Cartesian block to the far-field distance. This distance is usually around 100 mean aerodynamic chords, as recommended by the Third Drag Prediction Workshop [37]. The complete overset mesh of the SBW configuration contains 5.6 million cells.

2.4 Volume mesh deformation—pyWarp

We use a mesh deformation algorithm (pyWarp) to modify the initial volume meshes based on surface modifications dictated by the shape design variables at each optimization iteration. The pyWarp module performs the mesh deformation using an inexact explicit interpolation algorithm [38] to propagate the displacements and rotations of the surface nodes to the volume nodes. pyWarp requires updated positions of the surface mesh nodes as inputs. pyGeo provides the surface nodes of the primary components, while pySurf outputs the collar mesh surface nodes, as indicated in Fig. 5.

Each overset volume mesh is deformed independently under the influence of only its own surface nodes. For instance, the strut surface nodes only affect the deformation of the strut CFD mesh. This prevents the crushing of volume cells near the junction regions due to the simultaneous movement of primary components. The background mesh has no associated surface, and thus remains fixed throughout the optimization.

We apply Tapenade's reverse mode AD to the pyWarp module to compute the derivatives of the volume node positions with respect to the surface mesh changes.

2.5 Flow solution—ADflow

ADflow is a flow solver capable of solving Euler, laminar Navier-Stokes, and RANS equations in multiblock structured or overset meshes with a second-order cell-centered finite volume formulation. The inviscid fluxes are discretized with central-differencing and artificial dissipation, while the viscous fluxes use standard central-differencing. The code is based on the SUmb solver [39] and is augmented by implementing a Newton–Krylov solver, adjoint derivative computation [40], and overset mesh capability [17].

We use the SA-R-QCR2000 turbulence model for all flow simulations in this work, which is the standard SA turbulence model [41] augmented with rotation correction (R) [42] and quadratic constitutive relationship

(QCR2000) [43]. These corrections are necessary because the standard SA model overestimates trailing edge separation bubble sizes at junctions as the mesh is refined [44]. Furthermore, the standard SA model does not capture secondary recirculation structures at junction corners [45].

ADflow initially uses time-marching schemes, such as fourth-order Runge–Kutta or diagonalized diagonal dominant alternating direction implicit scheme [46], to approach the basin-of-attraction of the steady state. ADflow then switches to a Newton–Krylov algorithm to accelerate the convergence towards the steady solution [47]. The switching criterion is determined when time residuals drop below a user-specified threshold.

This CFD solver uses implicit hole cutting [19, 48] to determine the overset connectivities. This includes flagging which cells should be blanked, interpolated, or computed. We use first-order interpolation in the dual mesh to compute the interpolated cell values. We then utilize zipper meshes [49] to fill the surface gaps between overlapped meshes to provide a watertight surface for aerodynamic force integration. The overset connectivity is recomputed in every optimization iteration to take into account the relative displacements among the overset meshes. Kenway et al. [17] provide further details on the overset implementation in ADflow.

ADflow computes the sensitivities of the aerodynamic forces and moments with respect to the nodal coordinates with the hybrid adjoint approach [40, 50], in which we use Tapenade's reverse mode AD to compute the partial derivatives in the discrete adjoint equations. We use the preconditioned GMRES [51] within PETSc [52–54] to solve the resulting adjoint linear system.

The differentiated code does not consider the effect of the mesh coordinates on the overset interpolation weights. We solve this nonlinear relationship with an iterative Newton search during the forward execution of the code, and we are investigating appropriate ways to efficiently backpropagate derivatives across this step. Nevertheless, this assumption does not hinder the optimization [23] as we show in Sec. 4. The interpolation weights are updated in the next forward execution of the code.

2.6 Optimization—SNOPT

The analysis modules previously described provide function values and gradients to SNOPT (sparse nonlinear optimizer) [55], which is a gradient-based optimizer that implements the sequential quadratic programming method. SNOPT handles large-scale nonlinear optimization problems with thousands of constraints and design variables and is thus suitable for aerodynamic shape [56–58] and aerostructural optimizations [59–61].

We use pyOptSparse to handle the interface between SNOPT and the other modules of the MACH framework. pyOptSparse is an open source Python module that wraps several optimizers under a unified object-oriented interface. It is based on the original pyOpt framework [62]. pyOptSparse is available at `https://bitbucket.org/mdolab/pyoptsparse`.

3 Design optimization problem

The optimization objective in this work is to minimize the drag of the SBW for a given Mach number, lift coefficient, and wing planform. In other words, the optimizer may alter airfoil shapes and spanwise twist distributions, while the span and chords of the wing and strut remain fixed. The changes are applied over the whole wing and strut, including the wing-strut junction region.

We start with the baseline IGES description of the SBW provided in the PADRI 2017 website [24] to create structured surface meshes (Fig. 2) and triangulated surface meshes (Fig. 3) for each primary component. We then embed both surface meshes of each component into their corresponding FFD boxes (Fig. 4) to allow pyGeo to manipulate their shape. The fuselage has no associated FFD since its geometry remains fixed for this optimization. Regardless, we still require the fuselage triangulated surface for collar mesh generation. The shape variables consist of hundreds of control points distributed over the FFD boxes. This leads to the general formulation of the optimization problem in Table 2.

The wing FFD is comprised of two layers of 16 chordwise by 20 spanwise control points, with one layer for the upper surface and one for the lower. These control points move vertically to change the underlying wing surface shape. Control points within the same chordwise station can also be simultaneously rotated around the quarter-chord to change the local twist. Wing twist is controlled in 10 spanwise locations along the FFD and the intermediate twist values are interpolated between these locations. To avoid shear twist, the pairs

Table 2: SBW aerodynamic shape optimization problem.

	Variable/function	Description	Quantity
minimize	C_D	Drag coefficient	
with respect to	$-1.0 \leq \alpha \leq 3.0$	Angle of attack [deg]	1
	$-5.0 \leq \tau_{\text{wing}} \leq 5.0$	Wing sections twist [deg]	10
	$-0.05 \leq \Delta z_{\text{wing}} \leq 0.05$	Vertical displacement of wing FFD control points [m]	640
	$-10.0 \leq \tau_{\text{strut}} \leq 5.0$	Strut sections twist [deg]	11
	$-0.06 \leq \Delta n_{\text{strut}} \leq 0.06$	Normal displacement of strut FFD control points [m]	242
		Total design variables	**904**
subject to	$C_L = 0.4058$	Lift constraint (baseline at $\alpha = 1.0$ deg)	1
	$t/t_{\text{init,wing}} \geq 0.999$	Wing thickness constraint	60
	$\Delta z_{\text{wing,TE,upper}} = -\Delta z_{\text{wing,TE,lower}}$	Wing fixed trailing edge constraint	20
	$\Delta z_{\text{wing,LE,upper}} = -\Delta z_{\text{wing,LE,lower}}$	Wing fixed leading edge constraint	20
	$t/t_{\text{init,strut}} \geq 0.999$	Strut thickness constraint	84
	$\Delta n_{\text{strut,TE,upper}} = -\Delta n_{\text{strut,TE,lower}}$	Strut fixed trailing edge constraint	11
	$\Delta n_{\text{strut,LE,upper}} = -\Delta n_{\text{strut,LE,lower}}$	Strut fixed leading edge constraint	11
		Total constraints	**213**

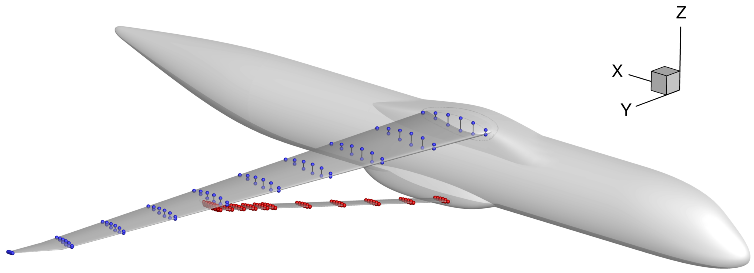


Figure 9: Points of the wing (blue) and strut (red) where thickness constraints are enforced.

of control points on the leading and trailing edges must move equal distances in opposing directions. This requirement is enforced by the leading and trailing edge constraints listed in Table 2.

The strut has control points defined for both the upper and lower surfaces, similar to the wing. The control points of each surface are distributed in a mesh of 11 chordwise by 11 spanwise control points. Each control point can be independently moved in the direction normal to the strut surface to change the thickness and camber distribution along the strut. In addition, 11 twist variables are defined for the strut sections (one for each chordwise section). We also enforce constraints on the leading and trailing edge control points to avoid shear twist of the strut sections. The aircraft angle of attack is an additional design variable to allow the optimizer to match the lift constraint.

Since structural analyses are not conducted during optimization, it is necessary to enforce artificial constraints to take structural considerations into account and to avoid unrealistically thin airfoil designs. For this optimization, we compute thicknesses in 144 locations along the wing and the strut, as shown in Fig. 9. The thicknesses at these points cannot go below their corresponding initial value.

The C_L constraint ensures that the optimized configuration generates the same lift as the baseline configuration at 1.0 degree of angle of attack. Preliminary aerodynamic analysis of the baseline configuration with ADflow shows that the lift coefficient value for this condition is $C_L = 0.4058$.

4 Optimization results

The solution of the SBW optimization takes 27 hours using 1088 threads distributed among 16 Intel Xeon Phi 7250 KNL nodes. The optimizer requires 69 CFD simulations and 40 adjoint solutions to arrive at the final result. The optimization history (Fig. 10) shows that the drag coefficient is reduced by 14.7%, while maintaining the lift coefficient of the initial configuration. The assumption of frozen interpolation weights in the reverse AD code does not compromise the optimization since the optimizer takes several unit steps, as indicated by the consecutive flow and adjoint evaluations in Fig. 10.

The optimality conditions are not satisfied to the specified tolerance, since the optimizer stops prematurely due to inconsistencies in the gradient caused by noise in the functions of interest. Changes in overset connectivities create discontinuities and noise in the design space, as shown in our previous work regarding a wing-fuselage junction optimization case [23]. This effect only becomes significant near the optimum, where variations caused by the noise exceed the shallow slopes of the underlying function. Despite this effect, the design has undergone significant improvement by this point of the optimization. Furthermore, during the final iterations we only observe minor changes to the geometry and the drag value decreases by thousandths of drag counts between iterations. Therefore, the shape can be considered optimal for practical purposes.

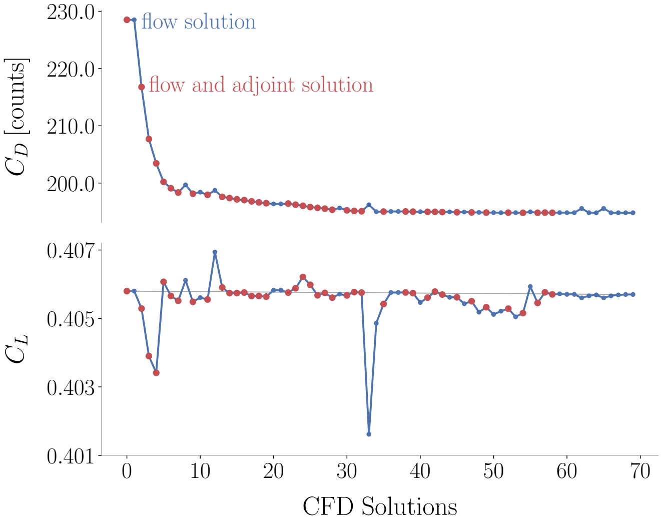


Figure 10: History of the SBW optimization. The drag decreases by 33 counts for the same lift coefficient.

The optimizer reduces the drag by improving the spanwise lift distribution, eliminating shock waves, and reducing the amount of separated flow. These correspond in turn to a reduction in induced drag, wave drag, and viscous drag, respectively.

The baseline configuration shows a shock wave near the wing-strut intersection region that is eliminated in the optimized design (see Fig. 11). The optimized shape reduces the flow acceleration in the gap between the two components by decreasing the strut incidence angle. Consequently, there is a pressure increase over the facing surfaces of the wing and the strut when compared to the baseline configuration (also shown in Fig. 11). The optimized solution also shows smoother pressure distributions, which slows boundary layer growth and thus decreases viscous drag.

The elimination of shocks has a positive impact on separation, as a large portion of separation observed in the baseline design is induced by shocks (Fig. 12). The wing-strut junction also induces separation on its trailing edge, and the optimizer manages to adjust the geometry of the intersection to reattach the flow in this region.

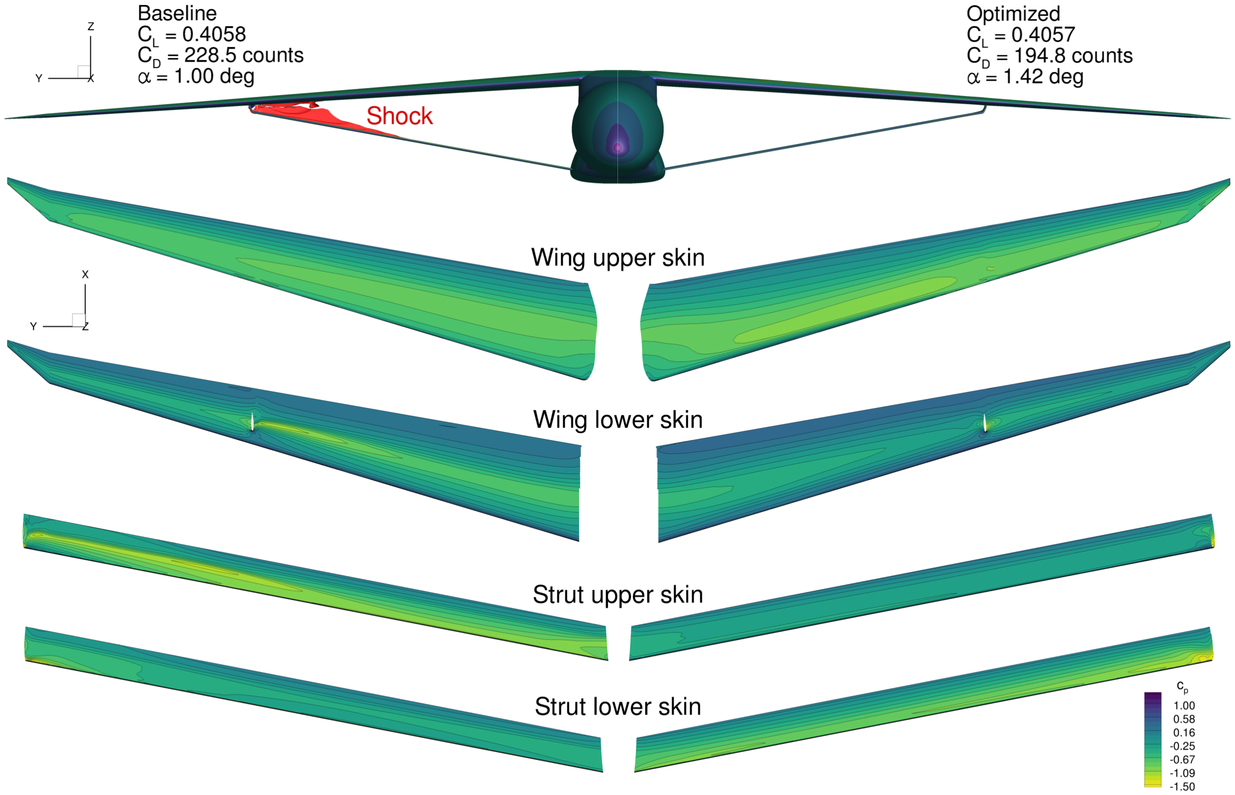


Figure 11: Comparison of shock and pressure distributions between the baseline (left) and optimized (right) designs. The strut is shown at a different scale.

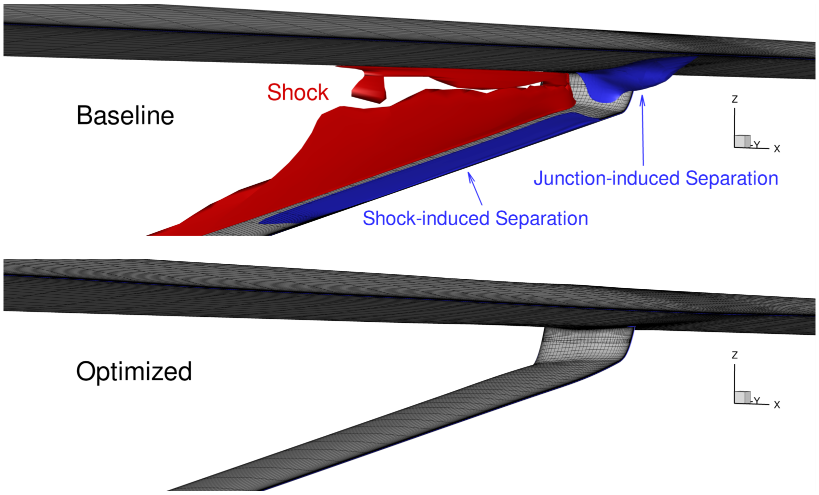


Figure 12: Separation on the trailing edge of the wing-strut junction before (top) and after (bottom) the optimization.

This study marks a notable improvement to our previous TBW investigation [16]. The previous optimization using multiblock meshes could not remove junction shocks and separation as effectively as the current approach based on overset meshes and component-based parametrization. In addition, the current work avoids the generation of negative volume cells that hindered the optimization in the original process.

While the previous work seemed to obtain a greater relative drag reduction, the results are not quantitatively comparable to this study as the previous optimization referenced a different baseline configuration, a different flight condition, and more permissive thickness constraints.

The incidence angle of the optimized strut is reduced to the point of generating negative lift, as shown in Fig. 13 and Fig. 14. This negative lift generation is compensated for by increased twist in the midspan part of the wing, resulting in an overall lift distribution that is close to elliptical. These trends were similarly observed in Euler- and RANS-based aerodynamic shape optimizations of different SBW geometries [14–16, 25]. Parametric studies done by Ko et al. [63] also show that using a strut with negative incidence angles and a relatively flat upper surface reduces the shock strength near the wing-strut intersection. Most of the drag reduction occurs from the inboard region of the wing and the strut sections near the junction, as shown in the upper right plot in Fig. 13.

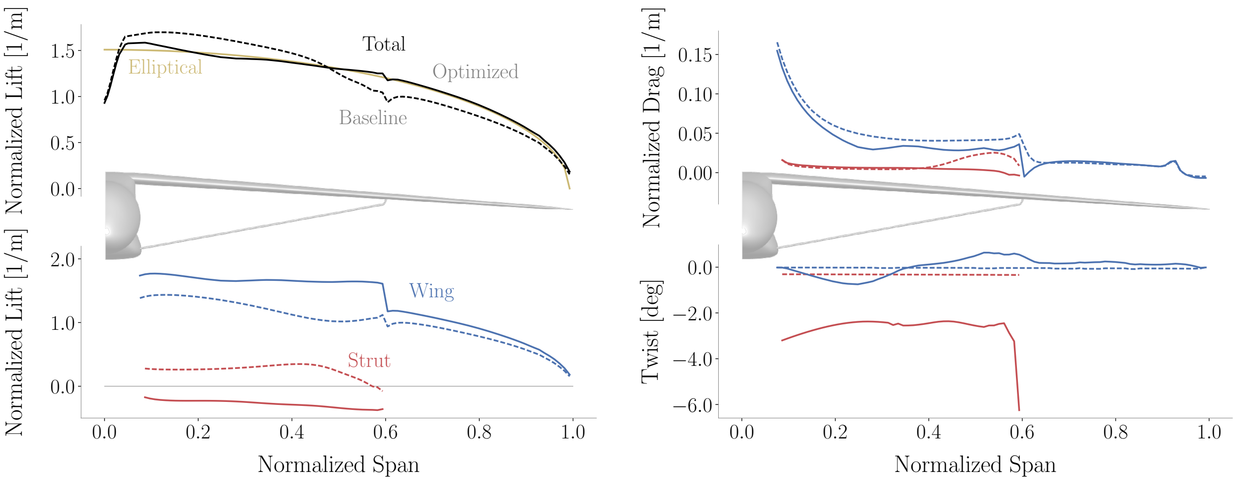


Figure 13: Spanwise distribution of lift, drag, and twist for the baseline (dashed lines) and optimized configuration (solid lines).

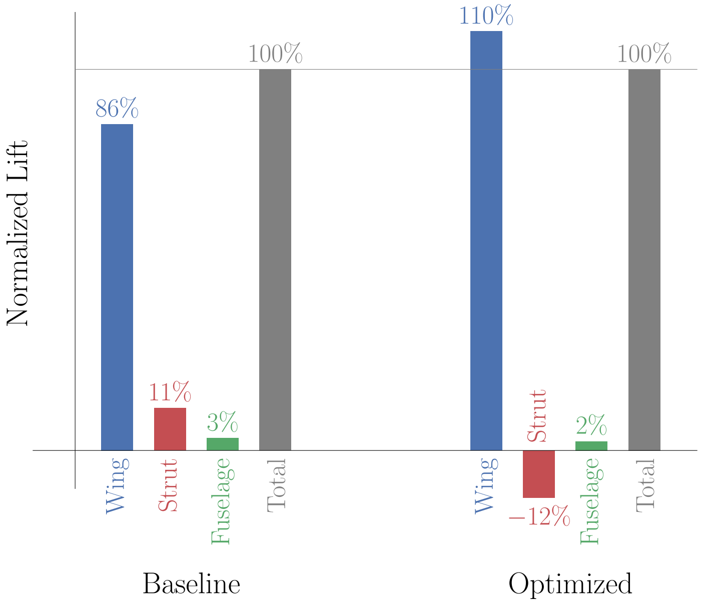


Figure 14: Component-wise lift distribution. The lift is normalized by the C_L constraint.

The optimization lowered the magnitude of the suction peaks on the lower surface of the wing and on the upper surface of the strut (Fig. 15). The wing slice at 59% of the semi-span of the optimized configuration has a subtle bump on the lower surface at approximately three quarters of the chord. The strut slice at 59% of the semi-span shows a bump near the trailing edge of the upper surface to increase the extent of favorable pressure gradient in this area, thus delaying separation. This trailing edge bump extends to the vertical section of the strut (Fig. 16). The redesigned shape eliminates the flat pressure distribution regions on the trailing edge caused by separation. The suction side of this vertical segment of the strut is on the outboard surface, following the same trend as the rest of the strut, to avoid high-speed flow on the inner side of the junction. According to the cross-sectional slices shown in Fig. 15 and Fig. 16, the optimization makes more significant changes to the strut rather than the wing in the quest to reduce drag.

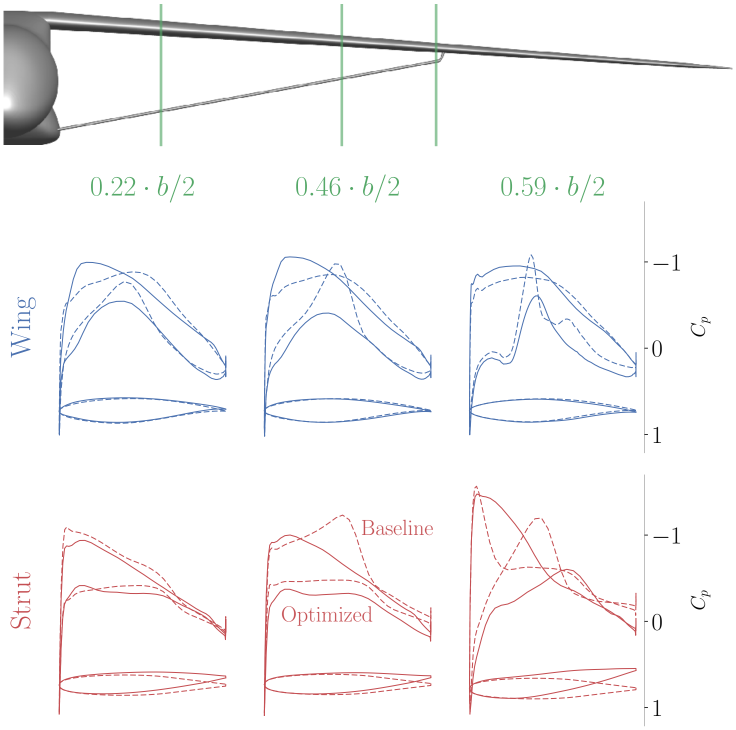


Figure 15: Cross-sectional slices of the wing and strut and corresponding pressure distributions for the baseline (dashed lines) and optimized (solid lines) configurations.

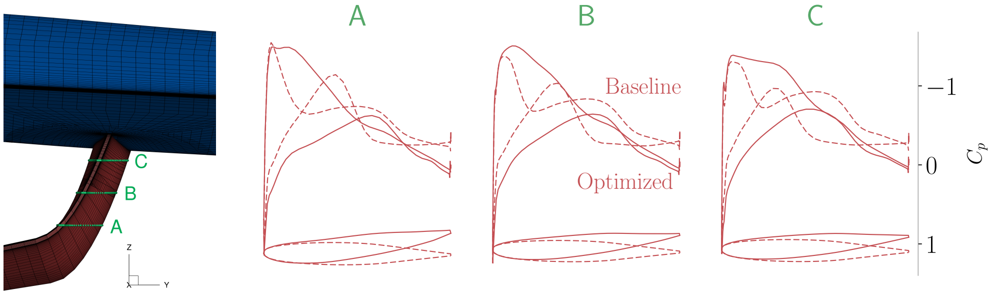


Figure 16: Cross-sectional slices of the vertical segment of the strut for the baseline (dashed lines) and optimized (solid lines) configurations.

5 Conclusion

In this work, we demonstrate the feasibility of RANS-based aerodynamic shape optimization of a transonic SBW geometry using overset meshes and component-based geometry manipulation. These methodologies avoid the mesh deformation issues seen in previous attempts and provide better shape control of the SBW components and their junctions.

The optimized design achieves a 14.7% drag reduction compared to the baseline configuration for the same lift. Studies of the optimized configuration show that a downward-lifting strut avoids high-speed flow in the wing-strut gap, eliminating the shock wave seen in the baseline configuration. To compensate for the negative lift of the strut and achieve an elliptical lift distribution overall, the load on the inboard wing section is increased. Adjustments to the wing-strut intersection curve and the addition of a bump on the strut trailing edge change the pressure recovery and reduce the separation. The optimization also demonstrates that changes to the strut shape are more effective for drag reduction rather than modifications to the wing shape.

These results highlight the fact that the use of high-fidelity analysis and optimization during the conceptual design of unconventional configurations facilitates the detection and solution of issues that might compromise their performance.

6 Acknowledgments

The authors acknowledge funding from the Brazilian Air Force and partial support by NASA through award number NNX14AC73A—Technical Monitor Tristan Hearn. This work used the Extreme Science and Engineering Discovery Environment (XSEDE), which is supported by National Science Foundation grant number ACI-1548562. We also acknowledge the help of John Jasa during the implementation of the pySurf module.

References

[1] Kenway, G. K. W., and Martins, J. R. R. A., "Multipoint High-Fidelity Aerostructural Optimization of a Transport Aircraft Configuration," *Journal of Aircraft*, Vol. 51, No. 1, 2014, pp. 144–160. doi:10.2514/1.C032150.

[2] Kennedy, G. J., Kenway, G. K. W., and Martins, J. R. R. A., "High Aspect Ratio Wing Design: Optimal Aerostructural Tradeoffs for the Next Generation of Materials," *Proceedings of the AIAA Science and Technology Forum and Exposition (SciTech)*, National Harbor, MD, 2014. doi:10.2514/6.2014-0596, AIAA 2014-0596.

[3] Gur, O., Bhatia, M., Schetz, J. A., Mason, W. H., Kapania, R. K., and Mavris, D. N., "Design Optimization of a Truss-Braced-Wing Transonic Transport Aircraft," *Journal of Aircraft*, Vol. 47, No. 6, 2010. doi:10.2514/1.47546.

[4] Chakraborty, I., Gross, J. R., Nam, T., Perullo, C., and Mavris, D. N., "Analysis of the Effect of Cruise Speed on Fuel Efficiency and Cost for a Truss-Braced Wing Concept," *14th AIAA Aviation Technology, Integration, and Operations Conference*, 2014. doi:10.2514/6.2014-2424, AIAA 2014-2424.

[5] Bradley, M. K., Droney, C. K., and Allen, T. J., "Subsonic Ultra Green Aircraft Research. Phase II-Volume I; Truss Braced Wing Design Exploration," *NASA Technical Report*, 2015. NASA/CR-2015-218704/VOL1.

[6] Carrier, G., Atinault, O., Dequand, S., Hantrais-Gervois, J., Liauzun, C., Paluch, B., Rodde, A., and Toussaint, C., "Investigation of a strut-braced wing configuration for future commercial transport," *28th Congress of the International Council of the Aeronautical Sciences (Brisbane, Australia)*, 2012.

[7] Turriziani, R., Lovell, W., Martin, G., Price, J., Swanson, E., and Washburn, G., "Preliminary design characteristics of a subsonic business jet concept employing an aspect ratio 25 strut braced wing," *NASA Technical Report*, 1980. NASA-CR-159361.

[8] Grasmeyer, J., "Multidisciplinary design optimization of a transonic strut-braced wing aircraft," *37th AIAA Aerospace Sciences Meeting and Exhibit*, 1999, pp. 11–14. doi:10.2514/6.1999-10.

[9] Gundlach, J. F., Tétrault, P.-A., Gern, F. H., Nagshineh-Pour, A. H., Ko, A., Schetz, J. A., Mason, W. H., Kapania, R. K., et al., "Conceptual design studies of a strut-braced wing transonic transport," *Journal of aircraft*, Vol. 37, No. 6, 2000, pp. 976–983. doi:10.2514/2.2724.

[10] Grasmeyer, J., and Mason, W., "A discrete vortex method for calculating the minimum induced drag and optimum load distribution for aircraft configurations with noncoplanar surfaces," *VPIAOE-242, AOE Department, VPI & SU, Blacksburg, Virginia*, Vol. 24061, 1997.

[11] Tétrault, P.-A., Schetz, J. A., and Grossman, B., "Numerical Prediction of Interference Drag of Strut-surface Intersection in Transonic Flow," *AIAA Journal*, Vol. 39, No. 5, 2001, pp. 857–864. doi:10.2514/2.1389.

[12] Duggirala, R. K., Roy, C. J., and Schetz, J. A., "Analysis of Interference Drag for Strut-strut Interaction in Transonic Flow," *AIAA Journal*, Vol. 49, No. 3, 2011, pp. 449–462. doi:10.2514/1.45703.

[13] Gipson, L., "NASA Aeronautics Budget Proposes Return of X-Planes," `http://www.nasa.gov/feature/nasa-aeronautics-budget-proposes-return-of-x-planes`, 2016-02-18. Accessed: 2016-04-22.

[14] Gagnon, H., and Zingg, D. W., "High-fidelity Aerodynamic Shape Optimization of Unconventional Aircraft through Axial Deformation," *52nd Aerospace Sciences Meeting*, 2014. doi:10.2514/6.2014-0908, AIAA 2014-0908.

[15] Gagnon, H., and Zingg, D. W., "Euler-equation-based drag minimization of unconventional aircraft configurations," *Journal of Aircraft*, Vol. 53, No. 5, 2016, pp. 1361–1371. doi:10.2514/1.C033591.

[16] Ivaldi, D., Secco, N. R., Chen, S., Hwang, J. T., and Martins, J. R. R. A., "Aerodynamic Shape Optimization of a Truss-Braced-Wing Aircraft," *Proceedings of the 16th AIAA/ISSMO Multidisciplinary Analysis and Optimization Conference*, Dallas, TX, 2015. doi:10.2514/6.2015-3436, AIAA 2015-3436.

[17] Kenway, G. K. W., Secco, N., Martins, J. R. R. A., Mishra, A., and Duraisamy, K., "An Efficient Parallel Overset Method for Aerodynamic Shape Optimization," *Proceedings of the 58th AIAA/ASCE/AHS/ASC Structures, Structural Dynamics, and Materials Conference, AIAA SciTech Forum*, Grapevine, TX, 2017. doi:10.2514/6.2017-0357, AIAA 2017-0357.

[18] Chan, W. M., and Steger, J. L., "Enhancements of a three-dimensional hyperbolic grid generation scheme," *Applied Mathematics and Computation*, Vol. 51, No. 2, 1992, pp. 181–205. doi:10.1016/0096-3003(92)90073-A.

[19] Landmann, B., and Montagnac, M., "A highly automated parallel Chimera method for overset grids based on the implicit hole cutting technique," *International Journal for Numerical Methods in Fluids*, Vol. 66, No. 6, 2011, pp. 778–804. doi:10.1002/fld.2292.

[20] Liao, W., and Tsai, H. M., "Aerodynamic Shape Optimization on Overset Grids Using the Adjoint Method," *International Journal for Numerical Methods in Fluids*, Vol. 62, No. 12, 2010, pp. 1332–1356. doi:10.1002/fld.2070.

[21] Lee, B. J., and Kim, C., "Aerodynamic redesign using discrete adjoint approach on overset mesh system," *Journal of Aircraft*, Vol. 45, No. 5, 2008, pp. 1643–1653. doi:10.2514/1.34112.

[22] Lee, B. J., Liou, M.-S., and Kim, C., "Optimizing a boundary-layer-ingestion offset inlet by discrete adjoint approach," *AIAA Journal*, Vol. 48, No. 9, 2010, pp. 2008–2016. doi:10.2514/1.J050222.

[23] Secco, N. R., Jasa, J. P., Kenway, G. K. W., and Martins, J. R. R. A., "Component-based Geometry Manipulation for Aerodynamic Shape Optimization with Overset Meshes," *18th AIAA/ISSMO Multidisciplinary Analysis and Optimization Conference*, American Institute of Aeronautics and Astronautics, 2017. doi:10.2514/6.2017-3327, AIAA 2017-3327.

[24] Bieler, H., Bier, N., Bugeda, G., Periaux, J., Redondo, D., Guttila, S., and Pons, J., "A common platform for validation of aircraft drag reduction technologies," , 2017. URL `http://congress.cimne.com/padri-2017/frontal/default.asp`.

[25] Hwang, J. T., Kenway, G. K. W., and Martins, J. R. R. A., "Geometry and Structural Modeling for High-Fidelity Aircraft Conceptual Design Optimization," *Proceedings of the 15th AIAA Multidisciplinary Analysis and Optimization Conference*, Atlanta, GA, 2014. doi:10.2514/6.2014-2041, AIAA 2014-2041.

[26] Meadows, N. A., Schetz, J. A., Kapania, R. K., Bhatia, M., and Seber, G., "Multidisciplinary design optimization of medium-range transonic truss-braced wing transport aircraft," *Journal of Aircraft*, Vol. 49, No. 6, 2012, pp. 1844–1856. doi:10.2514/1.C031695.

[27] Peter, J. E. V., and Dwight, R. P., "Numerical Sensitivity Analysis for Aerodynamic Optimization: A Survey of Approaches," *Computers and Fluids*, Vol. 39, No. 3, 2010, pp. 373–391. doi:10.1016/j.compfluid.2009.09.013.

[28] Kenway, G. K. W., Kennedy, G. J., and Martins, J. R. R. A., "Scalable Parallel Approach for High-Fidelity Steady-State Aeroelastic Analysis and Derivative Computations," *AIAA Journal*, Vol. 52, No. 5, 2014, pp. 935–951. doi:10.2514/1.J052255.

[29] Lambe, A. B., and Martins, J. R. R. A., "Extensions to the Design Structure Matrix for the Description of Multidisciplinary Design, Analysis, and Optimization Processes," *Structural and Multidisciplinary Optimization*, Vol. 46, 2012, pp. 273–284. doi:10.1007/s00158-012-0763-y.

[30] Kenway, G. K., Kennedy, G. J., and Martins, J. R. R. A., "A CAD-Free Approach to High-Fidelity Aerostructural Optimization," *Proceedings of the 13th AIAA/ISSMO Multidisciplinary Analysis Optimization Conference*, Fort Worth, TX, 2010. doi:10.2514/6.2010-9231, AIAA 2010-9231.

[31] Samareh, J. A., "Survey of Shape Parameterization Techniques for High-Fidelity Multidisciplinary Shape Optimization," *AIAA Journal*, Vol. 39, No. 5, 2001, pp. 877–884. doi:10.2514/1.C031695.

[32] Chan, W., Gomez, R., Rogers, S., and Buning, P., "Best practices in overset grid generation," *32nd AIAA Fluid Dynamics Conference and Exhibit*, 2002. doi:10.2514/6.2002-3191, AIAA 2002-3191.

[33] Bonet, J., and Peraire, J., "An alternating digital tree (ADT) algorithm for 3D geometric searching and intersection problems," *International Journal for Numerical Methods in Engineering*, Vol. 31, No. 1, 1991, pp. 1–17. doi:10.1002/nme.1620310102.

[34] Möller, T., "A fast triangle-triangle intersection test," *Journal of Graphics Tools*, Vol. 2, No. 2, 1997, pp. 25–30. doi:10.1080/10867651.1997.10487472.

[35] Chan, W. M., and Buning, P. G., "Surface grid generation methods for overset grids," *Computers & fluids*, Vol. 24, No. 5, 1995, pp. 509–522. doi:10.1016/0045-7930(95)00003-U.

[36] Hascoet, L., and Pascual, V., "The Tapenade automatic differentiation tool: Principles, model, and specification," *ACM Transactions on Mathematical Software*, Vol. 39, No. 3, 2013, pp. 20:1–20:43. doi:http://dx.doi.org/10.1145/2450153.2450158, URL `http://doi.acm.org/http://dx.doi.org/10.1145/2450153.2450158`.

[37] Vassberg, J., Tinoco, E., Mani, M., Brodersen, O., Eisfeld, B., Wahls, R., Morrison, J., Zickuhr, T., Laflin, K., and Mavriplis, D., "Summary of the third AIAA CFD drag prediction workshop," *45th AIAA Aerospace Sciences Meeting and Exhibit*, 2007. doi:10.2514/6.2007-260, AIAA 2007-260.

[38] Luke, E., Collins, E., and Blades, E., “A Fast Mesh Deformation Method Using Explicit Interpolation,” *Journal of Computational Physics*, Vol. 231, No. 2, 2012, pp. 586–601. doi:10.1016/j.jcp.2011.09.021.

[39] 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. doi:10.2514/6.2006-421, AIAA 2006-0421.

[40] 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, 2013. doi:10.2514/6.2013-2581, AIAA 2013-2581.

[41] Spalart, P. R., and Allmaras, S. R., “A one-equation turbulence model for aerodynamic flows,” *30th Aerospace Sciences Meeting and Exhibit*, 1992. doi:10.2514/6.1992-439, AIAA 1992-439.

[42] Dacles-Mariani, J., Zilliac, G. G., Chow, J. S., and Bradshaw, P., “Numerical/experimental study of a wingtip vortex in the near field,” *AIAA Journal*, Vol. 33, No. 9, 1995, pp. 1561–1568. doi:10.2514/3.12826.

[43] Spalart, P. R., “Strategies for turbulence modelling and simulations,” *International Journal of Heat and Fluid Flow*, Vol. 21, No. 3, 2000, pp. 252–263. doi:10.1016/S0142-727X(00)00007-2.

[44] Coder, J. G., Pulliam, T. H., Hue, D., Kenway, G. K., and Sclafani, A. J., “Contributions to the 6th AIAA CFD Drag Prediction Workshop Using Structured Grid Methods,” *55th AIAA Aerospace Sciences Meeting*, Grapevine, TX, 2017. doi:10.2514/6.2017-0960, AIAA 2017-0960.

[45] Jespersen, D. C., Pulliam, T. H., and Childs, M. L., “OVERFLOW Turbulence Modeling Resource Validation Results,” NAS Technical Report: NAS-2016-01, 2016. URL `https://turbmodels.larc.nasa.gov/NAS_Technical_Report_NAS-2016-01.pdf`.

[46] Klopfer, G., Hung, C., Van der Wijngaart, R., and Onufer, J., “A diagonalized diagonal dominant alternating direction implicit (D3ADI) scheme and subiteration correction,” *29th AIAA Fluid Dynamics Conference*, 1998. doi:10.2514/6.1998-2824, AIAA 1998-2824.

[47] Knoll, D. A., and Keyes, D. E., “Jacobian-free Newton–Krylov methods: a survey of approaches and applications,” *Journal of Computational Physics*, Vol. 193, No. 2, 2004, pp. 357–397. doi:10.1016/j.jcp.2003.08.010.

[48] Lee, Y., and Baeder, J. D., “Implicit hole cutting - a new approach to overset grid connectivity,” *16th AIAA Computational Fluid Dynamics Conference, Fluid Dynamics and Co-located Conferences*, 2003. doi:10.2514/6.2003-4128, AIAA 2003-4128.

[49] Chan, W. M., “Enhancements to the Hybrid Mesh Approach to Surface Loads Integration on Overset Structured Grids,” *19th AIAA Computational Fluid Dynamics, Fluid Dynamics and Co-located Conferences*, 2009. doi:10.2514/6.2009-3990, AIAA 2009-3990.

[50] 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, 2008, pp. 863–873. doi:10.2514/1.29123.

[51] Saad, Y., and Schultz, M. H., “GMRES: A Generalized Minimal Residual Algorithm for Solving Nonsymmetric Linear Systems,” *SIAM Journal on Scientific and Statistical Computing*, Vol. 7, No. 3, 1986, pp. 856–869. doi:10.1137/0907058.

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

[53] Balay, S., Buschelman, K., Gropp, W. D., Kaushik, D., Knepley, M. G., McInnes, L. C., Smith, B. F., and Zhang, H., "PETSc Web page," , 2009. Http://www.mcs.anl.gov/petsc.

[54] Balay, S., Abhyankar, S., Adams, M. F., Brown, J., Brune, P., Buschelman, K., Dalcin, L., Eijkhout, V., Gropp, W. D., Kaushik, D., Knepley, M. G., McInnes, L. C., Rupp, K., Smith, B. F., Zampini, S., Zhang, H., and Zhang, H., "PETSc Users Manual," Tech. Rep. ANL-95/11 - Revision 3.7, Argonne National Laboratory, 2016. URL http://www.mcs.anl.gov/petsc.

[55] Gill, P. E., Murray, W., and Saunders, M. A., "An SQP algorithm for large-scale constrained optimization," *Society for Industrial and Applied Mathematics*, Vol. 47, No. 1, 2005. doi:10.1137/S0036144504446096, URL http://www.stanford.edu/group/SOL/papers/SNOPT-SIGEST.pdf.

[56] Liem, R. P., Martins, J. R. R. A., and Kenway, G. K., "Expected Drag Minimization for Aerodynamic Design Optimization Based on Aircraft Operational Data," *Aerospace Science and Technology*, Vol. 63, 2017, pp. 344–362. doi:10.1016/j.ast.2017.01.006.

[57] Kenway, G. K. W., and Martins, J. R. R. A., "Buffet Onset Constraint Formulation for Aerodynamic Shape Optimization," *AIAA Journal*, Vol. 55, No. 6, 2017, pp. 1930–1947. doi:10.2514/1.J055172.

[58] Lyu, Z., Kenway, G. K. W., and Martins, J. R. R. A., "Aerodynamic Shape Optimization Investigations of the Common Research Model Wing Benchmark," *AIAA Journal*, Vol. 53, No. 4, 2015, pp. 968–985. doi:10.2514/1.J053318.

[59] Brooks, T. R., Kennedy, G. J., and Martins, J. R. R. A., "High-fidelity Multipoint Aerostructural Optimization of a High Aspect Ratio Tow-steered Composite Wing," *Proceedings of the 58th AIAA/ASCE/AHS/ASC Structures, Structural Dynamics, and Materials Conference, AIAA SciTech Forum*, Grapevine, TX, 2017. doi:10.2514/6.2017-1350, AIAA 2017-1350.

[60] Mader, C. A., Kenway, G. K., Martins, J., and Uranga, A., "Aerostructural Optimization of the D8 Wing with Varying Cruise Mach Numbers," *18th AIAA/ISSMO Multidisciplinary Analysis and Optimization Conference*, 2017. doi:10.2514/6.2017-4436, AIAA 2017-4436.

[61] Martins, J. R. R. A., Kenway, G. K. W., and Brooks, T. R., "Multidisciplinary Design Optimization of Aircraft Configurations—Part 2: High-fidelity aerostructural optimization," Lecture series, Von Karman Institute for Fluid Dynamics, Rode Saint Genèse, Belgium, May 2016. ISSN0377-8312.

[62] Perez, R. E., Jansen, P. W., and Martins, J. R. R. A., "pyOpt: A Python-Based Object-Oriented Framework for Nonlinear Constrained Optimization," *Structural and Multidisciplinary Optimization*, Vol. 45, No. 1, 2012, pp. 101–118. doi:10.1007/s00158-011-0666-3.

[63] Ko, A., Mason, W., and Grossman, B., "Transonic aerodynamics of a wing/pylon/strut juncture," *21st AIAA Applied Aerodynamics Conference*, 2003. doi:10.2514/6.2003-4062, AIAA 2003-4062.