



AERODYNAMIC SHAPE OPTIMIZATION OF FILLETED INTERSECTIONS WITH SURFACE MESH DEFORMATION

Hannah M. Hajdik¹, Anil Yildirim¹ & Joaquim R. R. A. Martins¹

¹University of Michigan

Abstract

Component intersections are a persistent challenge in aerodynamic design with the drag of a full configuration that is typically larger than that of its individual components. To remedy this, designers create fairings and fillets between the component surfaces to reduce this interference drag. These additions require iterative design and costly analyses to perfect, but aerodynamic shape optimization can find a better design quicker than traditional processes. For designs involving intersecting components, special handling of deformations is needed to preserve mesh quality within aerodynamic shape optimization. In this work, we present a method that uses inverse distance warping to deform a fillet between two components for the minimum drag design that retains mesh quality. We show that this method is tractable within an aerodynamic shape optimization process and results in lower drag designs than those optimized without changing the intersection region. This method enables better aircraft designs through more detailed aerodynamic shape optimization.

Keywords: aerodynamic shape optimization, mesh warping, component intersections, aircraft design

Introduction

Intersecting aerodynamic surfaces are a source of interference drag as both surfaces' boundary layers interact. Individual surfaces meeting at junctions result in drag higher than that of the components on their own, categorized as interference drag [1, Chapter 8]. This problem necessitates fairings, fillets, or other aerodynamic surfaces to prevent separation by making the surface more continuous. These surfaces require significant design and testing efforts to create [2], but testing does prove their benefit [3]. The design of these surfaces relies on highly specialized designer experience and intuition rather than methodical, well-understood design processes. Additionally, the testing procedure can be lengthy, making this highly iterative process both cost- and time-intensive. Creating these intersection regions alongside the rest of the design rather than as a late addition would result in a lower-drag design because design tradeoffs are done earlier on.

Aerodynamic shape optimization can fill the gap left by the lack of comprehensive design methods. When used with efficient gradient computation, aerodynamic shape optimization can find better designs than traditional iterative design processes, even when handling hundreds of design variables [4]. There are still challenges with aerodynamic shape optimization. Among other considerations, care must be taken to preserve the mesh quality for the computational fluid dynamics (CFD) solver throughout the optimization when deforming the mesh as the geometry changes rather than remeshing the geometry. The extra difficulty accounting for mesh deformation is necessary because remeshing hinders the optimization convergence by introducing noise [5].

Hadjilias [6] developed a method to design a fillet between a wing and fuselage at the preliminary design stage. This process used a surrogate modeling approach with a design of experiments (DOE) for the optimization, which was quick but limited to six design variables, all of which controlled the fillet.

Additionally, the CFD solver used to build the surrogate was a thin-layer Navier–Stokes approximation limited to small meshes due to computational cost. The fillet designed with this method was able to dramatically reduce interference drag despite the limited design freedom and CFD capability. Song and Lv [7] optimized a wing-body intersection with a two-part fairing. The B-spline approach they used was flexible enough for controlling the intersection, but the combination of a surrogate model with a DOE mean that only fourteen design variables could be considered. All design variables considered were on the fairing sections instead of also designing the wing and fuselage.

Yildirim et al. [8] optimized a fairing around intersecting components with a method that ensured the mesh would stay topologically consistent and only go through smooth changes. This used gradient-based optimization rather than a surrogate modeling approach. This method offers full manipulation of the fairing, but does not allow for C_1 continuous intersections. Chau and Zingg [9] optimized a strut-braced wing with gradient-based aerodynamic shape optimization where the wing-strut junction was handled separately in the parameterization. This method allows for the design of a junction with improved aerodynamic performance, but it does not allow for changes to be isolated to just one component. Changes made to the wing can affect the strut and vice versa.

Some of these methods are limited in the number of design variables they can handle and the exploration of the design space due to the use of surrogate models. This lack of freedom restricts the gains that can be achieved with optimization. Others are limited in the types of intermediate surfaces that they can design. There is no method that has efficient gradients and allows for the modification of fillet-style surface between independent components.

In this work, we introduce a method that can be used in gradient-based optimization of filleted surface intersections. Design changes to a fillet surface can be achieved while keeping the surface mesh consistent by using changes to the other components to influence the fillet mesh. By using the inverse distance method for mesh warping, points on the curves between a fillet and its neighbors are deformed by the neighbors and the changes are propagated inward. The fillet then follows along smoothly as the other components change, keeping the mesh topology in place. This method accomplishes two things: it deforms the CFD surface mesh to keep it topologically consistent, and it indirectly applies design changes to the fillet region. These indirect design changes are easier to set up and easier to compute in the optimization.

Without this method separately accounting for the fillet mesh deformation at the intersection region, two issues can arise when optimizing the wing and fuselage with free-form deformation (FFD) volumes [10]. Deformations from one FFD can seep over to points embedded in the other FFD. Conflicting changes to the intersection region from opposing FFD deformation can break the mesh topology.

Embedding the entire aircraft in one FFD can avoid the issue of conflicting changes because there is only one FFD, but this does not allow for independent fuselage and wing changes. We show this with one minimal FFD around just the wing-fuselage intersection (Figure 1) to mimic changes that would happen with a full parameterization.

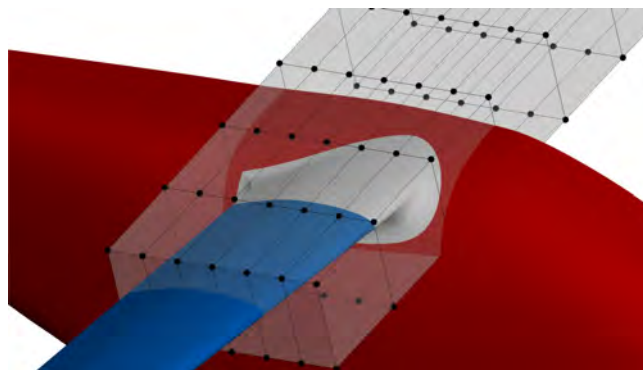
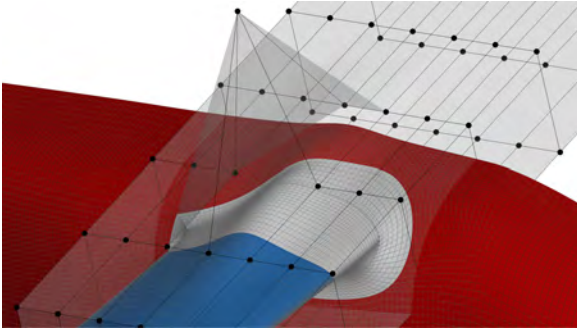
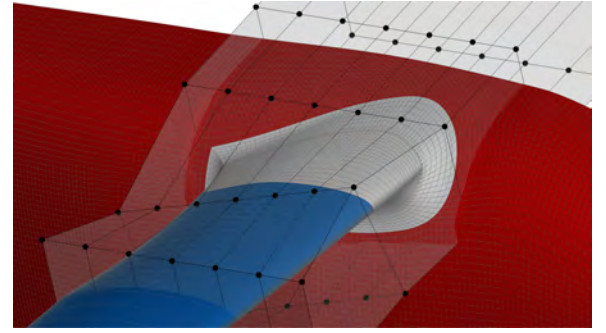


Figure 1 – The fuselage and wing can be embedded in one FFD.

In this case, all the mesh points are embedded in the same FFD, so deformations meant to be applied to the fuselage also affect nearby wing points and vice versa. This can be seen in Figure 2a where a perturbation to the fuselage shape also impacts the shape of the wing. In Figure 2b, the wing twist also moves the nearby fuselage points.



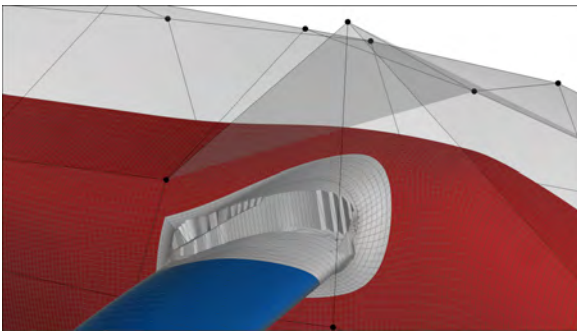
(a) A shape change on the fuselage can impact the wing.



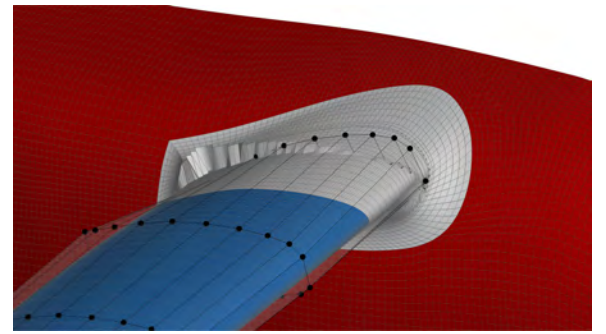
(b) A twist applied to the wing can deform the fuselage.

Figure 2 – Embedding the fuselage and wing in one FFD eliminates the ability to keep design changes confined to components.

Separating the parameterization into two FFDs (Figure 8) isolates the design changes. Without handling the intersection separately as in this method, this separation causes problems with mesh topology, shown in Figure 3. These issues arise from conflicts where both FFDs affect mesh points.



(a) A shape change on the fuselage can break the mesh at the intersection.



(b) A twist applied to the wing also breaks the mesh at the intersection region.

Figure 3 – If the aircraft is embedded in multiple FFDs, the mesh at the intersection region breaks without special methods to maintain its topology.

Method

The proposed method fits in to the geometry update and surface mesh deformation portion of the optimization process (Figure 4) of MACH [11], an open-source MDO framework. New design variables are set by the optimizer, SNOPT [12] used through pyOptSparse [13]. These design variables, in the form of FFD displacements [10], are applied to the surface mesh by pyGeo [14], which defines the geometric design variables and constraints. The new surface mesh points are used by IDWarp [15] to deform the volume mesh using the inverse distance method. ADflow [16], a structured multiblock solver with a Spallart–Allmaras turbulence model, solves the Reynolds-averaged Navier–Stokes (RANS) equations on the updated volume mesh. ADflow has an adjoint implementation that enables efficient gradient-based optimization [4].

In the geometry and surface mesh handling section, we handle the fillet separately from the other components. Here these are a wing and fuselage, but this method is general enough for any two intersecting components. The changes to the fillet propagate from its neighbors rather than coming directly from the application of FFD movement.

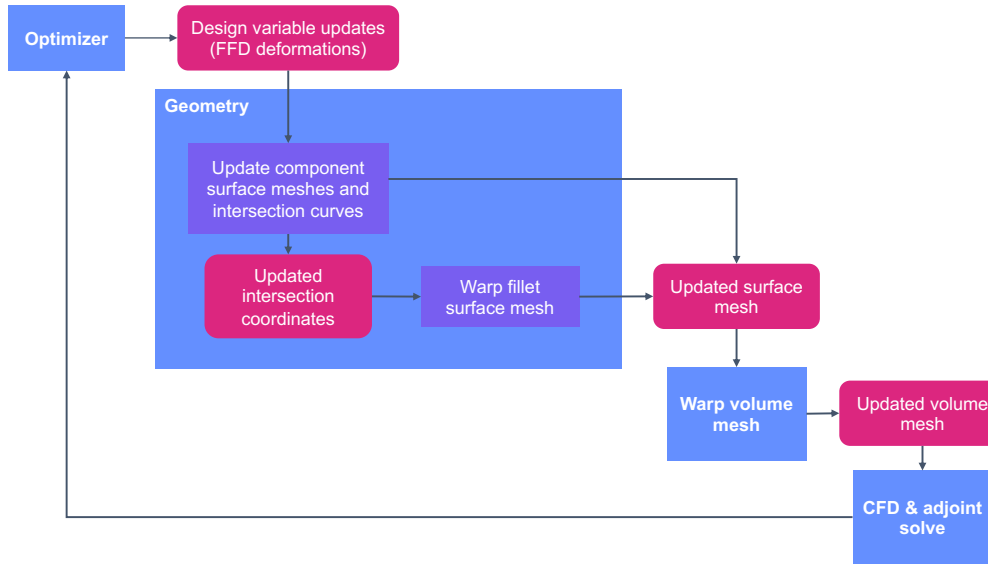


Figure 4 – The fillet is warped in the middle of the optimization process, replacing part of the normal surface mesh deformation.

A minimal example consisting of two components with two mesh points each, a “fillet” with three points, and essentially 2D design changes is shown in Figure 5. Each “curve” between the fillet and its neighboring components consists of a single overlapping point.

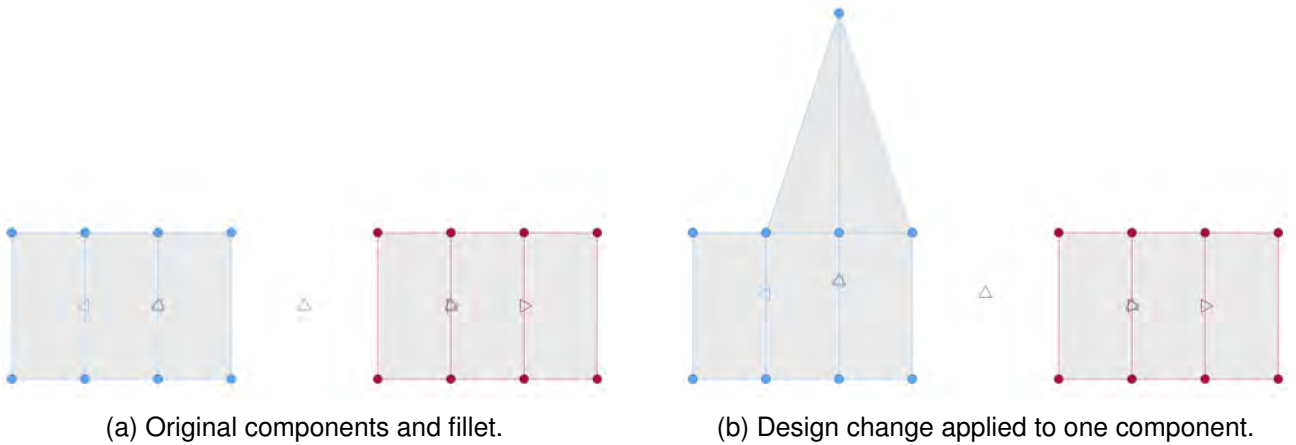


Figure 5 – Design changes to one component smoothly propagate through the intermediate region.

A representative configuration for this method has three pieces: two components (such as a wing and fuselage) and a fillet between them. Each must be composed of a continuous patch of the CFD surface mesh. Additionally, both components must be embedded in its own FFD with its own design variables, which keeps changes independent. The fillet is not embedded in any FFD and then has no changes applied through design variables, so all changes to the fillet result from changes to its neighbors.

Five pieces are tracked to enable this mesh deformation: each of the three pieces of the surface mesh mentioned above and the two curves between the fillet and the two components. The component surface meshes and the curves between them and the fillet move as the FFD volumes deform with changing design variables. The curve movements are then used as the “seeds” for the inverse distance warping scheme. Using these seeds, the entire fillet surface mesh is modified with a point-based inverse distance weighting scheme.

This scheme is a standard inverse distance method and the core functions described by Yildirim et al. [8]. This ensures the points on the boundary of the fillet and a component follow that component

exactly, keeping the mesh intact. Deformations to each point i of the fillet surface mesh, $\mathbf{x}_{\text{deform},i}$, are calculated using the displacements on each point j on the original curves between components after a design change, $\Delta \mathbf{p}_{S_0,j}$ and the weighting term $w_j(\mathbf{x}_{S_0,i})$ as follows:

$$\Delta \mathbf{x}_{\text{deform},i} = \frac{\sum_j w_j(\mathbf{x}_{S_0,i}) \Delta \mathbf{p}_{S_0,j}}{\sum_j w_j(\mathbf{x}_{S_0,i})}. \quad (1)$$

The weighting term is calculated from the distance between the fillet surface mesh point, $\mathbf{x}_{S_0,i}$, and the point on the original curve, $\mathbf{p}_{S_0,j}$. This distance is then inverted and raised to a power m as follows:

$$w_j(\mathbf{x}_{S_0,i}) = \frac{1}{\|\mathbf{x}_{S_0,i} - \mathbf{p}_{S_0,j}\|_2^m}. \quad (2)$$

We use $m = 3$ for simplicity, but more complex formulations exist.

When a design change is applied to one of the components as an FFD deformation, the fillet point on the pseudo-curve follows its matching point on the component exactly. The intermediate fillet point then deforms a fraction of the curve point. The fillet point on the opposite component does not move at all and neither do the points on that component.

One example of a design change made with this method can be seen in Figure 6. This shows a mounting angle change, but changes to mounting location, root airfoil shape, and fuselage shape near the fillet are also possible. Because the wing and fuselage mesh points are embedded in separate FFD volumes, changes to one component do not propagate to the other. The wing can twist without deforming the fuselage and the fuselage can also change in shape with no effect on the wing. The fillet is deformed in either case, but changes to its mesh originating from one component do not propagate to the other as in the case of a single FFD (Figure 2).

This method translates significant changes from the wing or fuselage geometry to the fillet between them. Changing the wing mounting angle up to 40° nose down is possible for this geometry and mesh (Figure 6a). This varies based on the particular geometry of the configuration and the mesh, especially at the fillet. This is demonstrated on this configuration because it can only twist 8° nose up (Figure 6b). The limiting factor is the volume mesh deformation, which needs the surface mesh to meet a certain quality to avoid negative volumes and other poor quality cells. Mesh quality is important for the accuracy and convergence of the CFD solution, and efficient mesh deformation is necessary for the speed of the optimization.

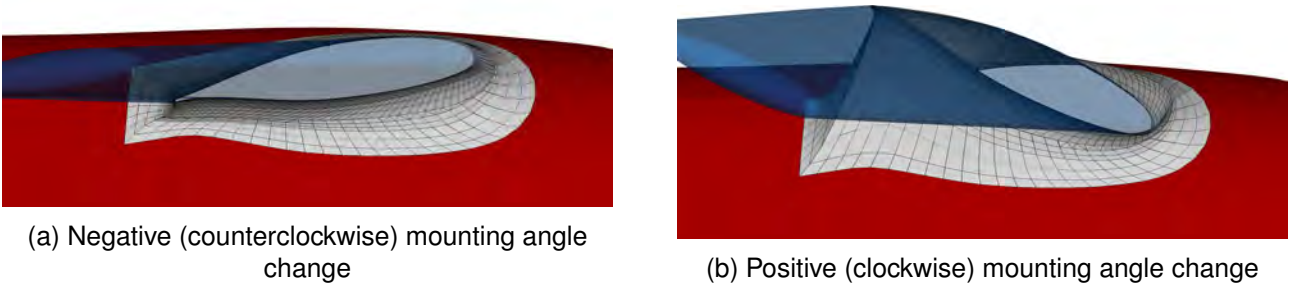


Figure 6 – The geometry changes that can be handled by this method depend on the geometry at hand and its effect on the volume mesh.

In the following sections, we will demonstrate the results of this method on the tiltwing concept aircraft. We compare a baseline optimization case where the fillet has no special treatment to an optimization where this method is used.

Aerodynamic shape optimization

Example aircraft configuration

This method is demonstrated on a conceptual tiltwing VTOL vehicle (Figure 7) developed by Whiteside et al. [17]. The fillet between the wing and fuselage is not in the original design and was initially added to make the meshing of the vehicle more tractable for previous studies. A similar fillet was added to this aircraft for CFD computations by Perez et al. [18].

In addition to the fillet, constant rotation is added to the wing to prevent negative volumes that consistently arose during optimizations. Using the fillet intersection warping method in this paper, the entire wing is rotated a constant -3° to deform the surface mesh of the wing and the fillet along with it. The new surface mesh is then re-extruded to form a new baseline. This is not possible without the proposed method but is used as the baseline for both optimization cases — with the moving fillet and without — for consistency.

This is not meant to be a representative study of how to improve this aircraft specifically because only the cruise wing is considered. The goal of this work is to demonstrate the fillet method, so we ignore the non-cruise conditions of this aircraft. We do not consider how the wing can be tilted 90° into its hover configuration with the addition of the fillet. The propellers and the tail are also not factored into this study.



Figure 7 – The VTOL concept vehicle by Whiteside et al. [17] in its cruise configuration.

Geometry parameterization

We show two optimizations on the concept tiltwing aircraft described in the previous section, one with this fillet method handling the intersection and one without it, to show the results of the method. Different FFD setups are necessary for the two cases, but the design variables and constraints are kept as consistent as possible between the two optimizations. The fuselage FFD is mirrored across the symmetry plane to preserve the continuity of the optimized shape across the symmetry plane. All fuselage design variables are symmetric across the symmetry plane.

The optimization with the fillet method uses the two-FFD setup shown in Figure 8. For the optimization without the fillet warping method, the only way to isolate design changes and not break the mesh is to prevent changes to the intersection region. This is accomplished with additional buffer layers in the FFD (Figure 9) that prevent design changes applied to one component from affecting another. This avoids the issues demonstrated previously in Figure 2 at the cost of not allowing changes to the intersection region.

The geometric design variables (DVs) and their bounds are listed in Table 1. Some design variables vary between the methods, but these differences are kept to a minimum to give both cases the same amount of design freedom to the extent their respective parameterizations will allow.

The wing twist is linear and rotates the wing sections about the quarter-chord. A root twist and a tip twist are applied and all intermediate sections are interpolated between those values. Because the optimization without the fillet method to handle the intersections cannot handle changes to the intersection region, the root twist is kept at zero. Outboard section twists are then determined in the same manner as the two-value twist. Enforcing a linear twist is done to prevent the optimizer from

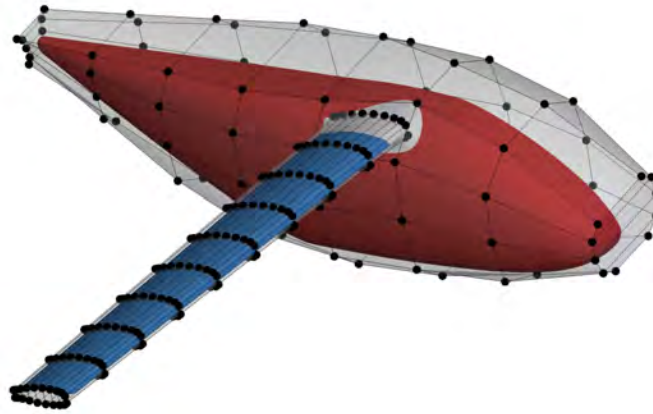


Figure 8 – Embedding the configuration in separate FFDs helps isolate design changes.

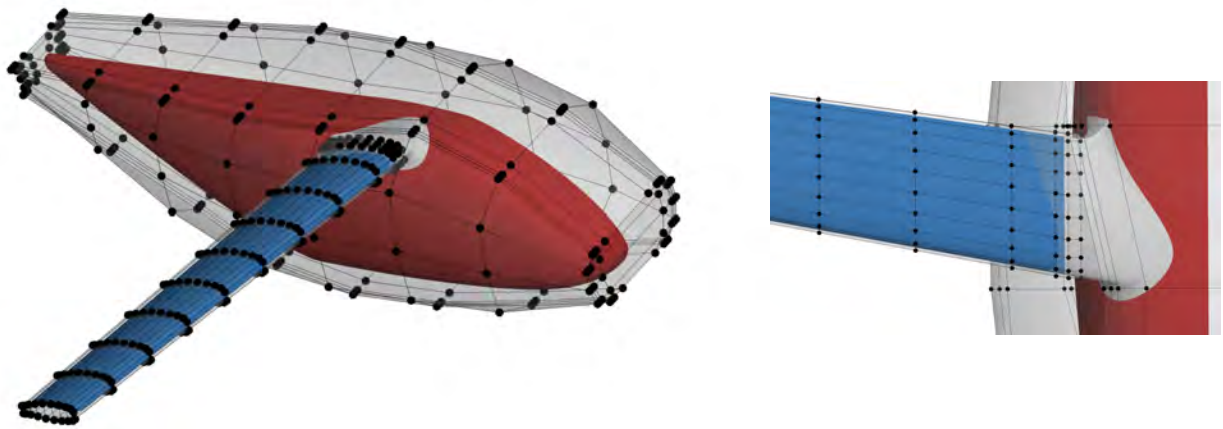


Figure 9 – Additional layers of control points (detail on right) act as a buffer region to prevent changes to other components and the intersection region.

Table 1 – Geometric design variables for both cases.

Design variable	n	Bounds	Scaling
Wing twist	1-2	$-10^\circ \leq x \leq -10^\circ$	10.0
Wing position*	1	$0.0m \leq x \leq 0.25m$	100.0
Wing shape	144-160	$-0.5m \leq x \leq 0.5m$	100.0
Fuselage shape	57	$-0.5m \leq x \leq 0.5m$	100.0
Non-fillet case total	202		
Fillet case total	220		

exploiting the single-point optimization [19].

Either 160 or 144 shape DVs are applied to the wing depending on whether the optimization is conducted with or without the fillet intersection method respectively. The wing FFD used for the fillet intersection method has an extra spanwise section of active FFD control points because of the space opened up by the lack of the three sections that act as a buffer for the fuselage.

The fuselage is parameterized with 57 shape DVs. The fuselage DVs move radially outward on the sides of the fuselage. On the upper and lower surfaces as well as on the front and back, the FFD control points can separately move vertically and horizontally. The four points on the fuselage around

fillet are more tightly bounded than the other control points to prevent mesh deformation failures.

A DV to move the wing outboard is also included. This shifts the root out without moving the tip, ensuring the span remains constant. The fuselage can move out over the root of the fillet, making this DV necessary to relieve pressure there and preserve the mesh quality. This DV is not used in the case without the fillet warping method because the minimal changes are made to the intersection region.

Identical geometric constraints are added to the wing and fuselage for both cases. These constraints act as a surrogate for the structure and internal components that are not explicitly modeled in these optimizations. A combination of thickness and volume constraints are used, with the bounds as listed in Table 2.

Table 2 – Geometric constraints for both optimization cases.

Constraint	n	Bounds	Scaling
Wing thickness	100	$x \geq \text{baseline}$	1.0
Wing volume	1	$x \geq \text{baseline}$	1.0
Fuselage thickness	27	$x \geq \text{baseline}$	1.0
Fuselage volume	3	$x \geq \text{baseline}$	1.0
Total	131		

CFD setup

The reference geometry, flow conditions, and mesh are kept constant between both aerodynamic shape optimizations (Table 3). The cruise conditions and reference values from Whiteside et al. [17] are used for the single design point in these optimizations. The initial value for angle of attack is set by solving for the value necessary to achieve the lift coefficient in Whiteside et al. [17]. The mesh used in these optimizations is an overset mesh with about 5.5 million compute cells. Both optimizations were run on 360 processors.

Table 3 – Geometry, flow condition, and mesh information for aerodynamic shape optimizations.

Reference area	5.946 m ²
Reference chord	0.967 m ²
Altitude	1219 m
Mach number	0.222
α	2.1861
Total cell count	5,879,632
Compute cell count	5,548,474

Optimization problem formulation

The optimization problem is kept as consistent as possible between the optimizations with and without the fillet warping intersection method (Table 4). Both optimizations have C_D as the objective. The geometric DVs do vary slightly as described in Table 1 but the geometric constraints (Table 2) as well as other DVs and constraints, the objective, and scaling are all consistent. Angle of attack also is included as a variable in both, resulting in final DV counts of 203 and 221 depending on the optimization.

The geometric constraints are constant between the two as outlined in Table 2. Both optimizations also have a constraint on C_L , the value for which comes from the cruise condition in Whiteside et al. [17]. The total number of constraints for both optimizations is then 132.

Table 4 – Optimization setup for aerodynamic shape optimizations.

		Quantity	Bounds	Scaling
Minimize	C_D	1		10^4
By varying	Geometric design variables	202-220		
	Angle of attack	1	$-10^\circ \leq x \leq 10^\circ$	0.1
Total, no intersection method		203		
Total, intersection method		221		
Subject to	Geometric constraints	131		
	C_L	1	$x = 0.873$	1.0
Total (both cases)		132		

Optimizations

We optimized the tiltwing concept aircraft both with and without this fillet warping method to obtain a baseline of what is possible without giving the intersection special treatment. When the intersection was locked instead of treating the fillet with this method, the reduction in drag of the optimized design was 11%. Using this method achieved a reduction in drag of 13%. Results from both optimizations are compared in Table 5. Using the fillet method to allow both the wing and fuselage to deform resulted in significant changes throughout the optimization without failures in the volume mesh warping routine.

Table 5 – Comparison of the baseline design and the two optimizations.

	AOA ($^\circ$)	C_L	C_D (counts)	$\Delta C_{D,baseline}$
Baseline	1.999	0.8728	434.30	
Optimization with locked intersection	6.644	0.8730	386.26	−11.0%
Optimization with fillet intersection method	7.719	0.8730	377.02	−13.2%

Compared to the optimization not using this method, the fillet method has the same lift constraint but has 9.24 fewer drag counts, a reduction of 2.4%. This can largely be attributed to the reduction in interference drag as wing and fuselage can be shaped in relation to each other more cohesively.

The angle of attack for the case using the intersection warping method to move the fillet is 1.08° higher than the case where the intersection is locked. This effect could be due to the difference in the wing twist between the cases. The optimized designs push strongly toward a very twisted wing for this aircraft in addition the already rotated baseline.

In the case without the fillet intersection, the twist of the wing root is kept at 0° (Table 1) to avoid breaking the mesh. In that case, the optimized value of the tip twist is -10° , which is at the lower bound, so all intermediate sections are interpolated between 0° and -10° . To twist the wing at the inboard sections, the only option the optimizer has is to dramatically increase the twist at the root. The root and tip twists can both change in the case using the fillet method, where the optimized values are -6.1° for the root and -9.3° for the tip. The optimizer then does not need to twist the wing as dramatically at the tip to change the root twist. This ability to twist the root of the wing affects the lift of the full configuration and the shape changes the optimizer needs to make to satisfy the constraints. The optimizer might be getting more lift from the centerbody of the aircraft when the root twist is favorable.

Based on the wing airfoil sections and their pressure plots (Figure 10), both optimizations follow a similar path in shaping the wing from the baseline. The similarities between the two are closer further outboard, as would be expected because that is where they have nearly the same control over the design. Both designs are closely aligned at the wingtip and alleviate the pressure peak

present in the baseline at the trailing edge of the tip. At the wing root, the design using the fillet warping method to control the intersection changed the airfoil much more significantly, both in twist and shape. The pressure distribution appears to worsen in the other design versus the baseline, but this is likely because the optimizer tries to achieve the lift constraint where it can despite the limitations in parameterization.

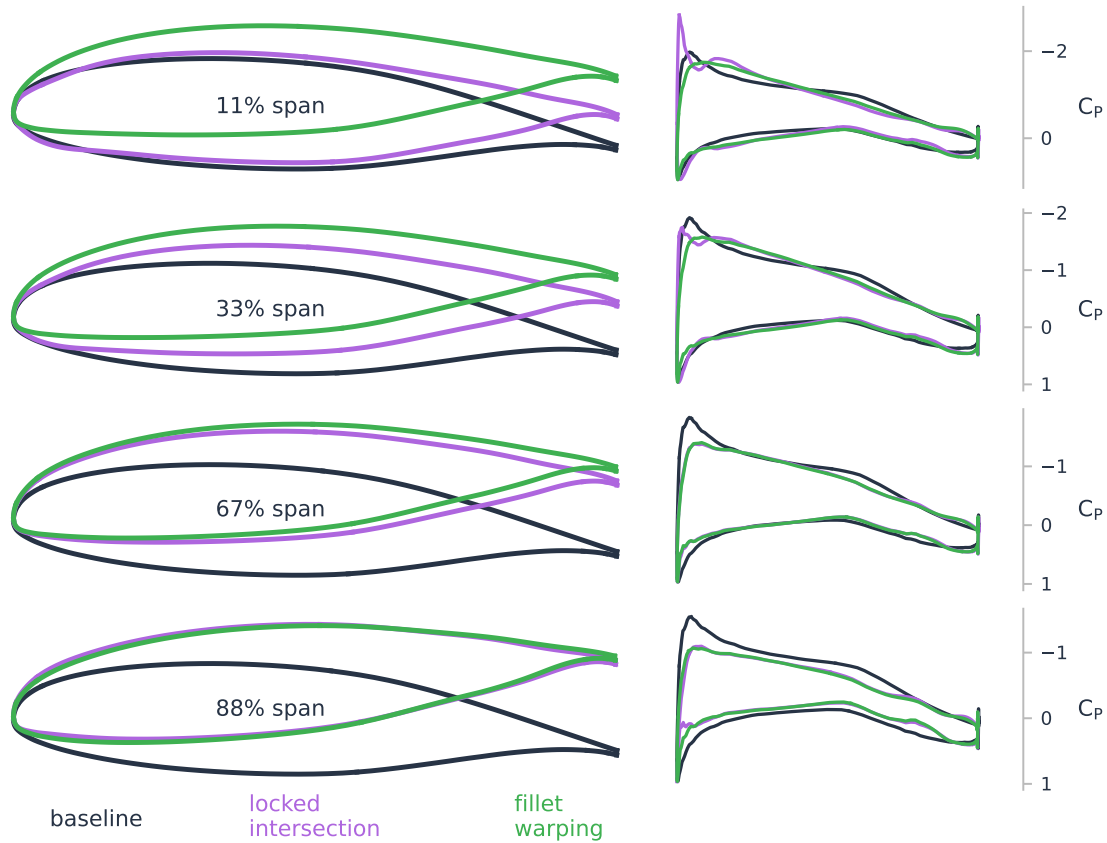


Figure 10 – Wing airfoil sections of the baseline design and optimized designs from both cases. Slices go from the wing-fillet boundary to the wingtip.

The airfoil sections and pressure plots of the fillet in Figure 11 highlight the effect of considering the fillet intersection in the optimization. In the optimization where the intersection is locked, the fillet still changes slightly, but only minor changes that filter through the buffer layers on the FFD volumes without breaking the mesh at the intersection. With the method to allow for warping the fillet, the airfoil shapes and twist change significantly as the wing and fuselage change. At the fuselage boundary, the fillet thins, reducing separation at the trailing edge and smoothing out the pressure distribution. The airfoils on both sides of the wing-fillet boundary have a noticeable flat profile on the lower surface, which appears to increase the lift of that section without an unfavorable drag increase.

The fuselage cross-sections in Figure 12 change similarly whether or not the fillet method is used in the optimization. The trends are generally toward slimming the fuselage, which makes sense because the geometric constraints are fairly permissive on the fuselage (Table 2). Because the geometric constraints are loose and only intended to keep the design within reasonable bounds rather than strictly capture practical considerations, the optimized designs are best used as a comparison against each other to determine how the separate parameterizations behave.

After the fillet (50% and 75% of the length), the design is much wider at the top than at the bottom in both optimized designs. This is likely because the optimizer uses the fuselage shape to help meet the lift constraint. This effect is also seen in the section before the fillet in the optimization that also moves the fillet, so there is some mechanism for either increasing lift with the new fillet design, decreasing the drag by shaping that section differently, or both.

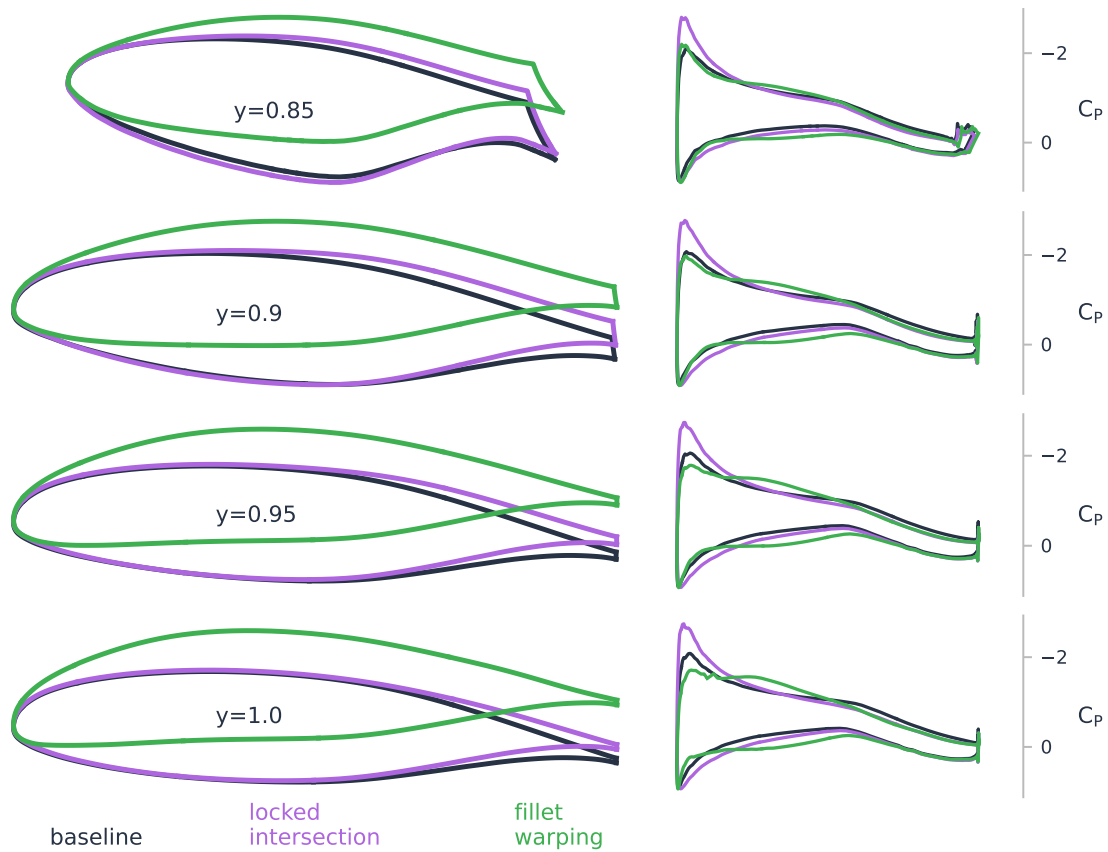


Figure 11 – Airfoil sections through the fillet on the baseline design and optimized designs from both cases from the fuselage-fillet boundary to the wing-fillet boundary.

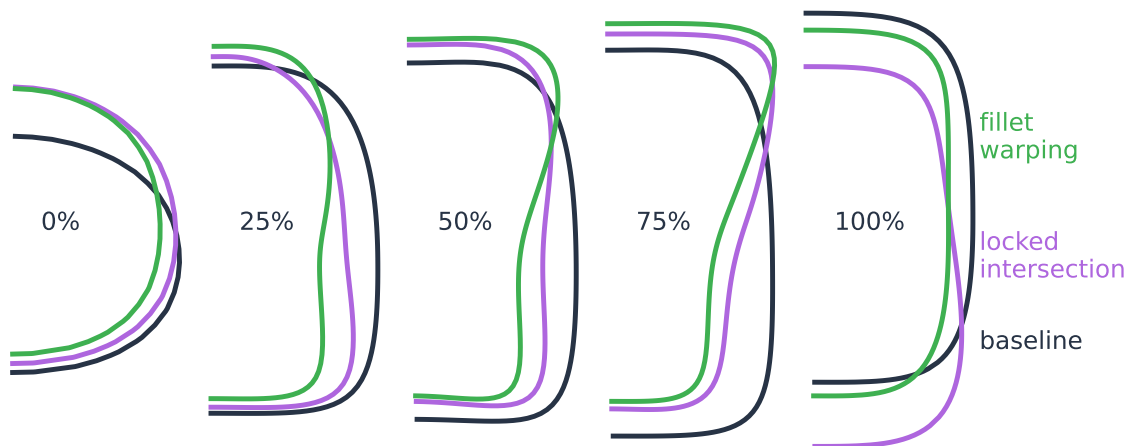


Figure 12 – Fuselage cross-sections on the baseline design and optimized designs from both cases from nose to the tail. The fillet is located from about 30%–45% of the length. Cross-sections are to scale with others at that station but not between stations to show detail on the smaller sections.

There is a large bump behind the wing of both optimized designs that can be seen in the surface contour plots in Figure 13 as well as the fuselage slices in Figure 12. There are a few possible explanations for this. The volume constraint imposed on the fuselage can be satisfied if, for example, the volume is taken from the sides and moved to the top, maintaining the same effective volume. The angles of attack for both optimized designs are quite high, so a prominent feature like this is not necessarily very prominent above the profile when it is at the specified angle of attack.

This region of the fuselage also has a significant region of low pressure and might help the aircraft meet the lift constraint. The bump is larger in the case where the intersection is locked and the optimization there struggles to meet the lift constraint because it cannot adjust the root twist, so it might need the additional help.

The wing position design variable was not used by the optimizer in the final design. This had benefits in some preliminary optimizations to relieve pressure on the fillet when the fuselage moved outward but was not necessary in this case.

The optimizations converged well in both cases (Figure 14). Both cases easily reached the specified feasibility requirement, satisfying the constraints to a tight tolerance. The optimization with the fillet method achieved one less order of magnitude of optimality but still converged to a tight tolerance. At the end of the optimization with the fillet method, the drag was changing on the order of 10^{-4} counts and at the end of the optimization without the drag was changing on the order of 10^{-5} counts. On 360 cores, the optimization with the fillet method took just under 30 hours to converge and the optimization with the method took just under 38, though that optimization was more complex due to the increase in design variables.

Conclusions

Interference drag from intersecting components is a crucial issue in aircraft design but can be mitigated through aerodynamic shape optimization with specialized methods. Aerodynamic shape optimization can improve the design of aerodynamic surfaces as long as it has a good geometric parameterization in place. By tracking the intersection curves between a fillet and the components on either side of it, those two components can be moved in relation to each other with the fillet following along smoothly. This allows for design variables such as mounting angle, root chord, and root position. These changes are only possible through the careful tracking of the intersection features.

We optimize a concept tiltwing aircraft both with and without this method handling the wing-fillet-fuselage intersection. The optimization is able to reduce the drag by an additional 9.6 counts, or 2.4%, with the method in place. Handling the intersection between components helps the optimizer find a design for the whole aircraft with even better performance than the components optimized on their own.

This method does not allow direct control of the fillet region and only enforces C^0 continuity on the intersection curves. Therefore, a non-tangent aerodynamic surface can be created. Even though this is likely less optimal than a surface with C^1 or greater continuity, the optimizer has no way to morph the geometry to this point. Additionally, the optimizations presented only consider one flow condition. To adequately design an aircraft using this method or any optimization, multiple flight points should be considered to capture the effect of the fillet at multiple relevant flow conditions.

We show that this method can make significant design changes to the wing and fuselage without breaking the mesh. With this method, it is possible to optimize a fillet between aerodynamic surfaces along with those surfaces. This allows for the design of better aircraft by enhancing existing methods for aerodynamic shape optimization. Comprehensive methods for designing aircraft surfaces are vital for the industrial adoption of aerodynamic shape optimization.

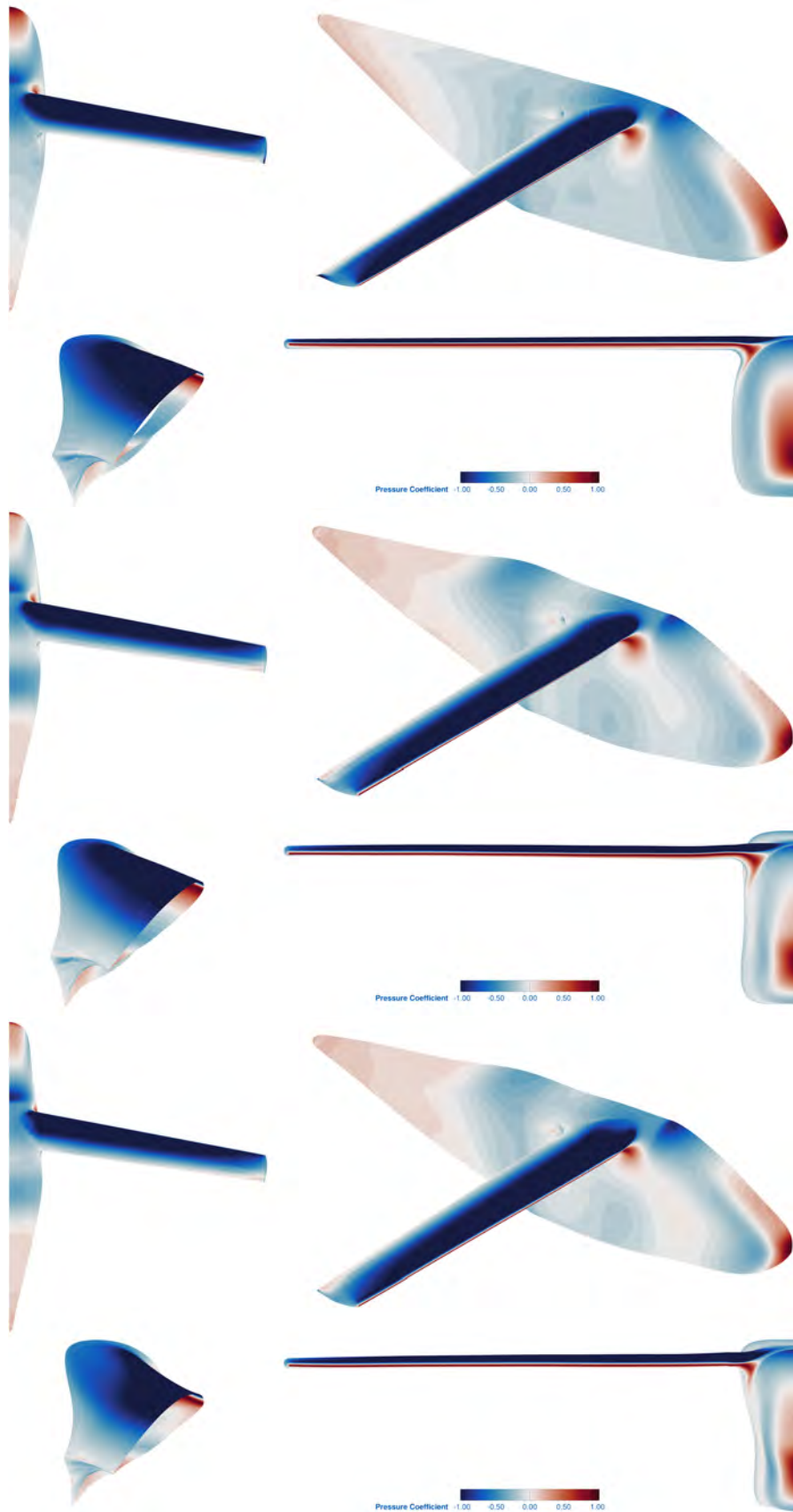


Figure 13 – Designs and pressure contours for: baseline, locked intersection optimization, and fillet warping optimization.

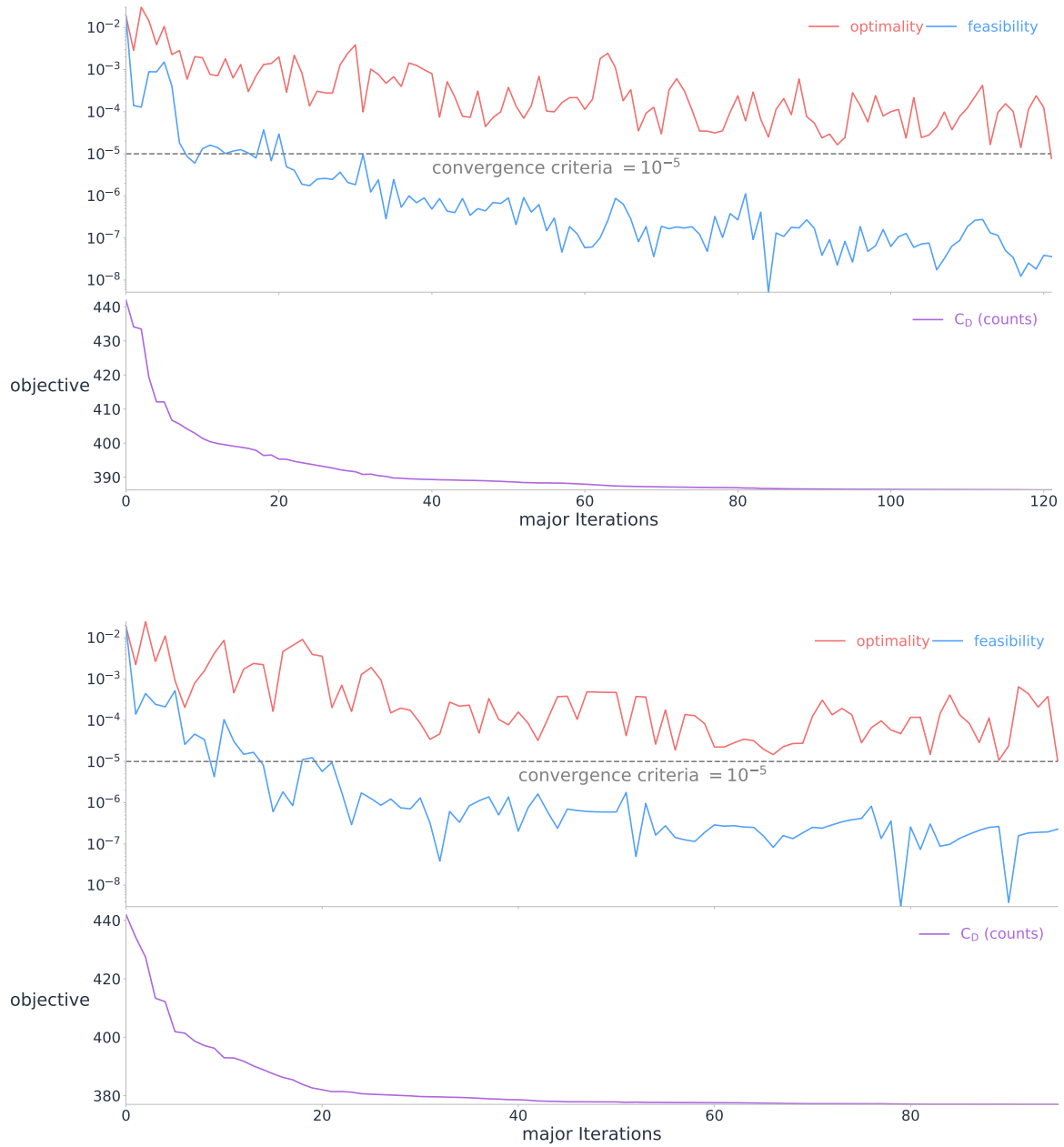


Figure 14 – Optimality, feasibility, and objective function history for locked intersection (top) and fillet method (bottom) optimizations.

Acknowledgements

We thank Sabet Seraj for his expert knowledge of MACH and Ali Gray and Eytan Adler for their plotting and debugging advice. H. Hajdik and J. R. R. A. Martins were partially supported by the U. S. Air Force Research Laboratory (AFRL) under the Michigan–AFRL Collaborative Center in Aerospace Vehicle Design (CCAVID) with Richard Snyder as the task Technical Monitor. Computational resources were provided by the NASA High-End Computing (HEC) Program through the NASA Advanced Supercomputing (NAS) Division at Ames Research Center.

Contact Author Email Address

The first author can be contacted at hajdik@umich.edu.

Copyright Statement

The authors confirm that they hold copyright on all of the original material included in this paper. The authors confirm that they give permission for the publication and distribution of this paper as part of the ICAS proceedings or as individual off-prints from the proceedings.

References

- [1] S. F. Hoerner. *Fluid-Dynamic Drag*. Hoerner Fluid Dynamics, Bakersfield, CA, 1965.
- [2] John Vassberg, Anthony Sclafani, and Mark DeHaan. A wing-body fairing design for the DLR-F6 model: A DPW-III case study. In *23rd AIAA Applied Aerodynamics Conference*, Fluid Dynamics and Co-located Conferences. American Institute of Aeronautics and Astronautics, June 2005. doi: 10.2514/6.2005-4730.
- [3] John Vassberg, Edward Tinoco, Mori Mani, Olaf Brodersen, Bernhard Eisfeld, Richard Wahls, Joseph Morrison, Tom Zickuhr, Kelly Laflin, and Dimitri Mavriplis. Summary of the third AIAA CFD drag prediction workshop. In *45th AIAA Aerospace Sciences Meeting and Exhibit*, Aerospace Sciences Meetings. American Institute of Aeronautics and Astronautics, January 2007. doi: 10.2514/6.2007-260.
- [4] Gaetan K. W. Kenway, Charles A. Mader, Ping He, and Joaquim R. R. A. Martins. Effective adjoint approaches for computational fluid dynamics. *Progress in Aerospace Sciences*, 110: 100542, October 2019. doi: 10.1016/j.paerosci.2019.05.002.
- [5] Ney Secco. *Component-Based Aerodynamic Shape Optimization using Overset Meshes*. PhD thesis, University of Michigan, Ann Arbor, MI, 2018.
- [6] Hippokrates A. Hadjiilias. *The aerodynamic design and optimization of a wing-fuselage junction fillet as part of a multi-disciplinary optimization process during the early aircraft design stages*. PhD thesis, Cranfield University, June 1996.
- [7] Wenbin Song and Peipei Lv. Two-level wing-body-fairing optimization of a civil transport aircraft. *Journal of Aircraft*, 48(6):2114–2121, 2011. doi: 10.2514/1.C031472.
- [8] Anil Yildirim, Charles A. Mader, and Joaquim R. R. A. Martins. A surface mesh deformation method near component intersections for high-fidelity design optimization. *Engineering with Computers*, January 2021. doi: 10.1007/s00366-020-01247-w.
- [9] Timothy Chau and David W. Zingg. Aerodynamic design optimization of a transonic strut-braced-wing regional aircraft. *Journal of Aircraft*, 59(1):253–271, 2022.
- [10] Thomas W. Sederberg and Scott R. Parry. Free-form deformation of solid geometric models. *SIGGRAPH Comput. Graph.*, 20(4):151–160, August 1986. ISSN 0097-8930. doi: 10.1145/15886.15903.

- [11] Gaetan K. W. Kenway, Graeme J. Kennedy, and Joaquim R. R. A. Martins. A scalable parallel approach for high-fidelity aerostructural analysis and optimization. In *53rd AIAA/ASME/ASCE/AHS/ASC Structures, Structural Dynamics, and Materials Conference*, Honolulu, HI, April 2012. AIAA 2012-1922.
- [12] Philip E. Gill, Walter Murray, and Michael A. Saunders. SNOPT: An SQP algorithm for large-scale constrained optimization. *SIAM Journal of Optimization*, 12(4):979–1006, 2002. doi: 10.1137/S1052623499350013.
- [13] Ella Wu, Gaetan Kenway, Charles A. Mader, John Jasa, and Joaquim R. R. A. Martins. py-OptSparse: a Python framework for large-scale constrained nonlinear optimization of sparse systems. *Journal of Open Source Software*, 5(54):2564, October 2020. doi: 10.21105/joss.02564.
- [14] Hannah M. Hajdik, Anil Yildirim, Ella Wu, Benjamin J. Brelje, Sabet Seraj, Marco Mangano, Joshua L. Anibal, Eirikur Jonsson, Eytan J. Adler, Charles A. Mader, Gaetan K. W. Kenway, and Joaquim R. R. A. Martins. pyGeo: A geometry package for multidisciplinary design optimization. *Journal of Open Source Software*, 8(87):5319, July 2023. doi: 10.21105/joss.05319.
- [15] Ney Secco, Gaetan K. W. Kenway, Ping He, Charles A. Mader, and Joaquim R. R. A. Martins. Efficient mesh generation and deformation for aerodynamic shape optimization. *AIAA Journal*, 59(4):1151–1168, April 2021. doi: 10.2514/1.J059491.
- [16] Charles A. Mader, Gaetan K. W. Kenway, Anil Yildirim, and Joaquim R. R. A. Martins. ADflow: An open-source computational fluid dynamics solver for aerodynamic and multidisciplinary optimization. *Journal of Aerospace Information Systems*, 17(9):508–527, September 2020. doi: 10.2514/1.I010796.
- [17] Siena K. S. Whiteside, Beau P. Pollard, Kevin R. Antcliff, Nikolas S. Zawodny, Xiaofan Fei, Christopher Silva, and Glenn L. Medina. Design of a tiltwing concept vehicle for urban air mobility. Technical Report NASA/TM-20210017971, NASA Langley Research Center, 2021.
- [18] David Garcia Perez, Patricia Ventura Diaz, and Steven Yoon. High-Fidelity Simulations of a Tiltwing Vehicle for Urban Air Mobility. In *AIAA SciTech Forum*, National Harbor, MD, 2023. American Institute of Aeronautics and Astronautics. doi: 10.2514/6.2023-2282.
- [19] Gaetan K. W. Kenway and Joaquim R. R. A. Martins. Multipoint aerodynamic shape optimization investigations of the Common Research Model wing. In *Proceedings of the AIAA Science and Technology Forum and Exposition (SciTech)*, Kissimmee, FL, January 2015. doi: 10.2514/6.2015-0264.