Chapter 1

Introduction

1.1 What is "MDO"?

Multidisciplinary design optimization (MDO) is the application of numerical optimization techniques to the design of engineering systems that involve multiple disciplines. Aircraft are prime examples of multidisciplinary systems, so it is no coincidence that MDO emerged within the aerospace community.

Before covering MDO, we first need to cover the "O", i.e. numerical optimization. This consists in the use of algorithms to minimize or maximize a given function by varying a number of variables. The problem might or not be subject to constraints.

Since many engineering design problems seek to maximize some measure of performance, it became obvious that using numerical optimization in design could be extremely useful, hence *design optimization* (the "DO" in MDO). Structural design was one of the first applications. A typical structural design optimization problem is to minimize the weight by varying structural thicknesses subject to stress constraints.

MDO emerged in the 1980s following the success of the application of numerical optimization techniques to structural design in the 1970s. Aircraft design was one of the first applications of MDO because there is much to be gained by the simultaneous consideration of the various disciplines involved (structures, aerodynamics, propulsion, stability and controls, etc.), which are tightly coupled. As an example, suppose that you are able to save one unit of weight in the structure. Because the coupled nature of all the aircraft weight dependencies, and the reduction in induced drag, the total reduction in the aircraft gross weight will be several times the structural weight reduction (about 5 for a typical airliner).

In spite of its usefulness, DO and MDO remain underused industry. There are several reasons for this, one of which is the absence of MDO in undergraduate and graduate curricula. This is changing, as most top aerospace departments nowadays include at least one graduate level course on MDO. We have also seen MDO being increasingly used by the major airframes.

The design optimization process follows a similar iterative procedure to that of the conventional design process, with a few key differences. Figure 1.1 illustrates this comparison. Both approaches must start with a baseline design derived from the specifications. This baseline design usually requires some engineering intuition and represents an initial idea. In the conventional design process this baseline design is analyzed in some way to determine its performance. This could involve numerical modeling or actual building and testing. The design is then evaluated based on the results and the designer then decides whether the design is good enough or not. If the answer is no — which is likely to be the case for at least the first few iterations — the designer will change the design based on his or her intuition, experience or trade studies. When the design is satisfactory, the designer will arrive at the final design.

For more complex engineering systems, there are multiple levels and thus cycles in the design process. In aircraft design, these would correspond to the preliminary, conceptual and detailed design stages.

The design optimization process can be pictured using the same flow chart, with modifications to some of the blocks. Instead of having the option to build a prototype, the analysis step must be completely numerical and must not involve any input from the designer. The evaluation of the design is strictly based on numerical values for the objective to be minimized and the constraints that need to be satisfied. When a rigorous optimization algorithm is used, the decision to finalize the design is made only when the current design satisfies the necessary optimality conditions that ensure that no other design "close by" is better. The changes in the design are made automatically by the optimization algorithm and do not require the intervention of the designer. On the other hand, the designer must decide in advance which parameters can be changed. In the design optimization process, it is crucial that the designer formulate the optimization problem well. We will now discuss the components of this formulation in more detail: the objective function, the constraints, and the design variables.

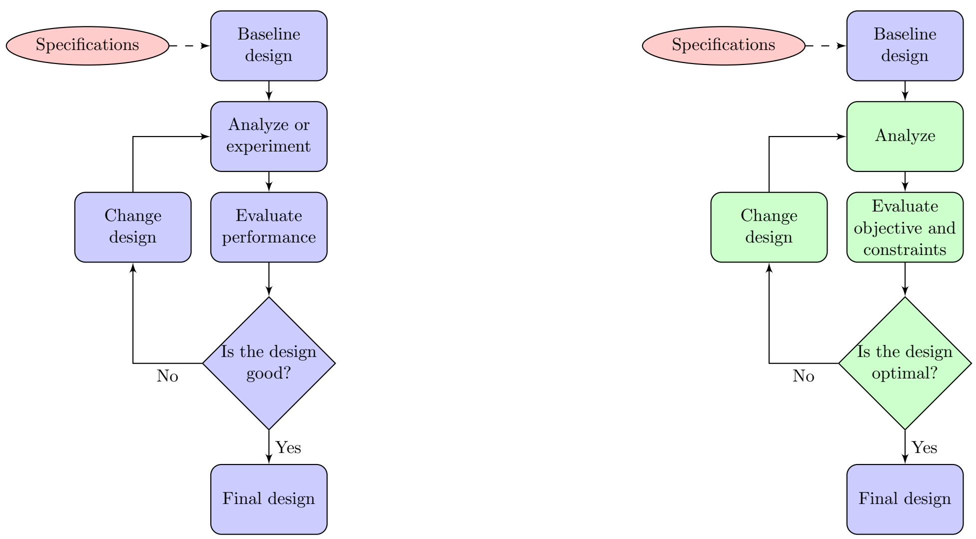


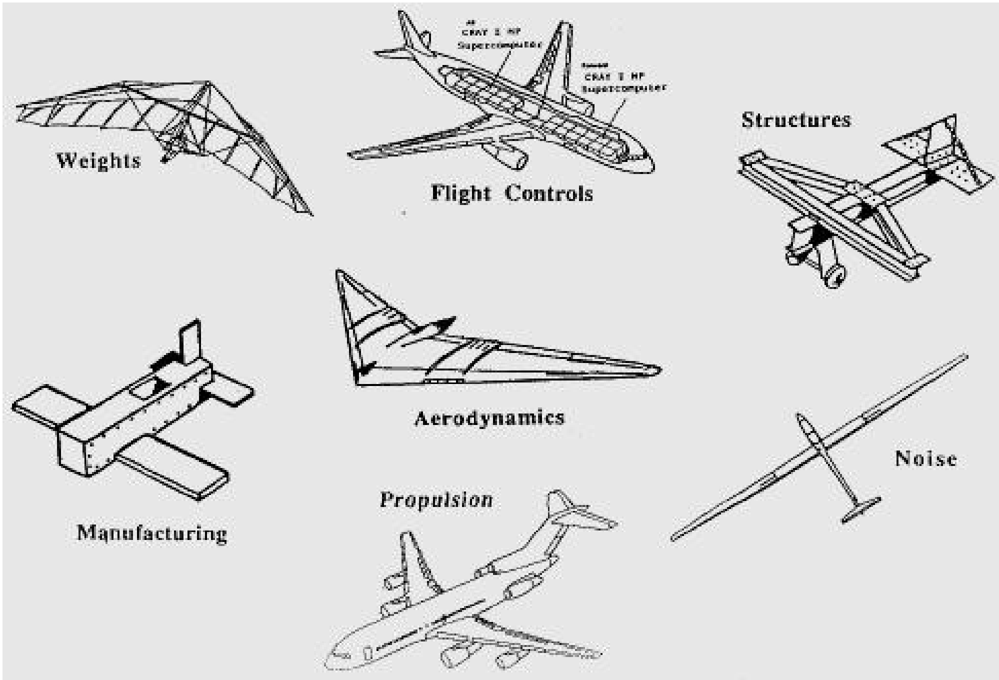
Figure 1.1: Conventional (left) versus optimal (right) design process

1.2 Terminology and Problem Statement

The formulation of the problem statement is often undervalued. Even experienced designers are sometimes not clear on what the different components in the optimization problem statement mean. Common conceptual errors include confusing constraints with objective functions. If we get the problem formulation wrong, then the solution of that problem could fail or simply return a mathematical optimum that is not the engineering optimum.

1.2.1 Objective Function

The objective function, f, is the scalar that we want to minimize. This is the convention in numerical optimization, so if we want to maximize a function, then we simply multiply it by -1



or invert it. This function thus represents a measure of "badness" that we want to minimize. Examples include weight in structural design and direct operating cost in aircraft design. To perform numerical optimization, we require the objective function to be computable for a range of designs. The objective function can be given as an explicit function, or be the result of a complex computational procedure, like the drag coefficient given by computational fluid dynamics.

The choice of objective function is crucial: if the wrong function is used, it does not matter how accurate the computation of the objective is, or how efficient the numerical optimization solves the problem. The "optimal" design is likely to be obviously non-optimal from the engineering point of view. A bad choice of objective function is a common mistake in MDO.

Optimization problems are classified with respect to the objective function by determining how the function depends on the design variables (linear, quadratic or generally nonlinear). In this course, we will concentrate on the general nonlinear case, although as we will see, there is merit in analyzing the quadratic case.

One common misconception is that in engineering design we often want to optimize multiple objectives. As we will se later, it is possible to consider problems with multiple objectives, but this results in a family of optimum designs with different emphasis on the various objectives. In most cases, it makes much more sense to convert these "objectives" into constraints. In the end, we can only make one thing best at the time.

The "Disciplanes"

Is there *one* aircraft which is the fastest, most efficient, quietest, most inexpensive?

"You can only make one thing best at a time."

1.2.2 Design Variables

The design variables, x, are the parameters that the optimization algorithm is free to choose in order to minimize the objective function. The optimization problem formulation allows for upper and lower *bounds* for each design variable (also known as *side constraints*). We distinguish these bounds from constraints, since as we will see later, they require a different numerical treatment when solving an optimization problem.

Design variables must be truly independent variables, i.e., in a given analysis of the design, they must be input parameters that remain fixed in the analysis cycle. They must not depend on each other or any other parameter.

Design variables can be broadly classified as quantitative and qualitative. Quantitative variables can be expressed by a number, which could be continuous or discrete. A real variable will result in the continuous case if allowed to vary at least within a certain range. If only discrete values are allowed (e.g., only plates of a certain thickness in a structure are available), then this results in the discrete case. Integer design variables will invariable result in the discrete case. Qualitative variables do not represent quantities, but discrete choices (e.g., the choice of aircraft configuration: monoplane vs. biplane, etc.).

In this course, we will deal exclusively with continuous design variables.

1.2.3 Constraints

The vast majority of practical design optimization problems have constraints. These are functions of the design variables that we want to restrict in some way. When we restrict a function to being equal to a fixed quantity, we call this an *equality constraint*. When the function is required to be greater or equal to a certain quantity, we have an *inequality constraint*. The convention we will use of inequality constraints is greater or equal, so we will have to make sure that less or equal constraints are multiplied by -1 before solving the problem. Note that some authors consider the less or equal constraints to be the convention.

As in the case of the objective function, the constraint functions can be linear, quadratic or nonlinear. We will focus on the nonlinear case, but the linear case will provide the basis for some of the optimization algorithms.

Inequality constraints can be be *active* or *inactive* at the optimum point. If inactive, then the corresponding constraint could have been left out of the problem with no change in its solution. In the general case, however, it is difficult to know in advance which constraints are active or not.

Constrained optimization is the subject of Chapter 5.

1.2.4 Optimization Problem Statement

Now that we have defined the objective function, design variables, and constraints, we can formalize the optimization problem statement as shown below.

$$
\begin{array}{rll}
\text{minimize} & f(x) & \\
\text{by varying} & x \in \mathbb{R}^n & \\
\text{subject to} & \hat{c}_j(x) = 0, & j = 1, 2, \ldots, \hat{m} \\
& c_k(x) \geq 0, & k = 1, 2, \ldots, m
\end{array}
\tag{1.1}
$$

f : objective function, output (e.g. structural weight).

x : vector of design variables, inputs (e.g. aerodynamic shape); bounds can be set on these variables.

$\hat{c}$: vector of equality constraints (e.g. lift); in general these are nonlinear functions of the design variables.

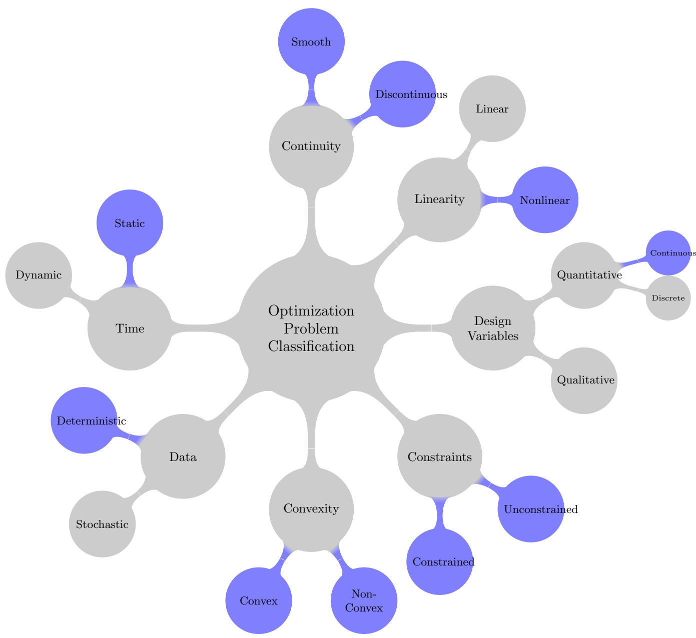


Figure 1.2: Classification of optimization problems; the course focuses on the types highlighted in blue.

c : vector of inequality constraints (e.g. structural stresses), may also be nonlinear and implicit.

where f is the objective function, x is the vector of n design variables, $\hat{c}$ is the vector of $\hat{m}$ equality constraints and c is the vector of m inequality constraints.

1.2.5 Classification of Optimization Problems

Optimization problems are classified based on the various characteristics of the objective function, constraint functions, and design variables. Figure 1.2 shows some of the possibilities.

In this course, we will consider the general nonlinear, non-convex constrained optimization problems with continuous design variables. However, you should be aware that restricting the optimization to certain types and using specialized optimization algorithms can result in a dramatic improvement in the capacity to solve a problem.

1.3 Timeline of Historical Developments in Optimization

300 bc: Euclid considers the minimal distance between a point a line, and proves that a square has the greatest area among the rectangles with given total length of edges.

200 bc: Zenodorus works on “Dido’s Problem”, which involved finding the figure bounded by a line that has the maximum area for a given perimeter.

100 bc: Heron proves that light travels between two points through the path with shortest length when reflecting from a mirror, resulting in an angle of reflection equal to the angle of incidence.

1615: Johannes Kepler finds the optimal dimensions of wine barrel. He also formulated an early version of the “marriage problem” (a classical application of dynamic programming also

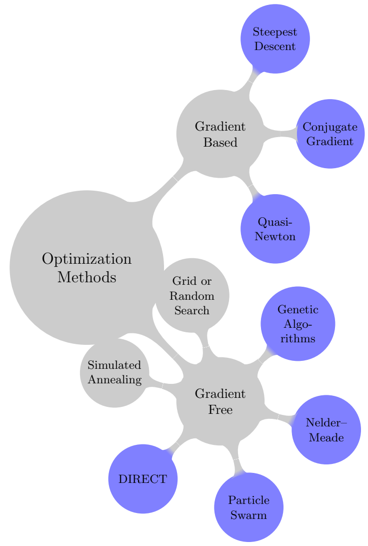


Figure 1.3: Optimization methods for nonlinear problems

known as the "secretary problem") when he started to look for his second wife. The problem involved maximizing a utility function based on the balance of virtues and drawbacks of 11 candidates.

1621 W. van Royen Snell discovers the law of refraction. This law follows the more general *principle of least time* (or Fermat's principle), which states that a ray of light going from one point to another will follow the path that takes the least time.

1646: P. de Fermat shows that the gradient of a function is zero at the extreme point the gradient of a function.

1695: Isaac Newton solves for the shape of a symmetrical body of revolution that minimizes fluid drag using calculus of variations.

1696: Johann Bernoulli challenges all the mathematicians in the world to find the path of a body subject to gravity that minimizes the travel time between two points of different heights — the *brachistochrone problem*. Bernoulli already had a solution that he kept secret. Five mathematicians respond with solutions: Isaac Newton, Jakob Bernoulli (Johann's brother), Gottfried Leibniz, Ehrenfried Walther von Tschirnhaus and Guillaume de l'Hôpital. Newton reportedly started solving the problem as soon as he received it, did not sleep that night and took almost 12 hours to solve it, sending back the solution that same day.

1740: L. Euler's publication begins the research on general theory of calculus of variations.

1746: P. L. Maupertuis proposed the *principle of least action*, which unifies various laws of physical motion. This is the precursor of the variation principle of stationary action, which uses calculus of variations and plays a central role in Lagrangian and Hamiltonian classical mechanics.

1784: G. Monge investigates a combinatorial optimization problem known as the *transportation problem*.

1805: Adrien Legendre describes the *method of least squares*, which was used in the prediction of asteroid orbits and curve fitting. Frederich Gauss publishes a rigorous mathematical foundation for the method of least squares and claims he used to predict the orbit of the asteroid Ceres in 1801. Legendre and Gauss engage in a bitter dispute on who first developed the method.

1815: D. Ricardo publishes the *law of diminishing returns* for land cultivation.

1847: A. L. Cauchy presents the steepest descent methods, the first gradient-based method.

1857: J. W. Gibbs shows that chemical equilibrium is attained when the energy is a minimum.

1902: Gyula Farkas presents and important lemma that is later used in the proof of the Karush–Kuhn–Tucker theorem.

1917: H. Hancock publishes the first text book on optimization.

1932: K. Menger presents a general formulation of the *traveling salesman problem*, one of the most intensively studied problems in optimization.

1939: William Karush derives the necessary conditions for the inequality constrained problem in his Masters thesis. Harold Kuhn and Albert Tucker rediscover these conditions an publish their seminal paper in 1951. These became known as the Karush–Kuhn–Tucker (KKT) conditions.

1939 Leonid Kantorovich develops a technique to solve linear optimization problems after having given the task of optimizing production in the Soviet government plywood industry.

1947: George Dantzig publishes the simplex algorithm. Dantzig, who worked for the US Air Force, reinvented and developed linear programming further to plan expenditures and returns in order to reduce costs to the army and increase losses to the enemy in World War II. The algorithm was kept secret until its publication.

1947: John von Neumann develops the theory of duality for linear problems.

1949: The first international conference on optimization, the International Symposium on Mathematical Programming, is held in Chicago.

1951: H. Markowitz presents his portfolio theory that is based on quadratic optimization. He receives the Nobel memorial prize in economics in 1990.

1954: L. R. Ford and D. R. Fulkerson research network problems, founding the field of combinatorial optimization.

1957: R. Bellman presents the necessary optimality conditions for dynamic programming problems. The Bellman equation was first applied to engineering control theory, and subsequently became an important principle in the development of economic theory.

1959: Davidon develops the first quasi-Newton method for solving nonlinear optimization problems. Fletcher and Powell publish further developments in 1963.

1960: Zoutendijk presents the methods of feasible directions to generalize the Simplex method for nonlinear programs. Rosen, Wolfe, and Powell develop similar ideas.

1963: Wilson invents the *sequential quadratic programming method* for the first time. Han reinvents it in 1975 and Powell does the same in 1977.

1975: Pironneau publishes the a seminal paper on aerodynamic shape optimization, which first proposes the use of adjoint methods for sensitivity analysis [12].

1975: John Holland proposed the first genetic algorithm.

1977: Raphael Haftka publishes one of the first multidisciplinary design optimization (MDO) applications, in a paper entitled "Optimization of flexible wing structures subject to strength and induced drag constraints" [2].

1979: Kachiyan proposes the first polynomial time algorithm for linear problems. The New York times publishes the front headline "A Soviet Discovery Rocks World of Mathematics", saying, "A surprise discovery by an obscure Soviet mathematician has rocked the world of mathematics and computer analysis . . . Apart from its profound theoretical interest, the new discovery may be applicable in weather prediction, complicated industrial processes, petroleum refining, t.he scheduling of workers at large factories . . . the theory of secret codes could eventually be affected by the Russian discovery, and this fact has obvious importance to intelligence agencies everywhere." In 1975, Kantorovich and T.C. Koopmans get the Nobel memorial price of economics for their contributions on linear programming.

1984: Narendra Karmarkar starts the age of interior point methods by proposing a more efficient algorithm for solving linear problems. In a particular application in communications network optimization, the solution time was reduced from weeks to days, enabling faster business and policy decisions. Karmarkar's algorithm stimulated the development of several other interior point methods, some of which are used in current codes for solving linear programs.

1985: The first conference in MDO, the Multidisciplinary Analysis and Optimization (MA&O) conference, takes place.

1988: Jameson develops adjoint-based aerodynamic shape optimization for computational fluid dynamics (CFD).

1995: Kennedy and Eberhart propose the particle swarm optimization algorithm.

1.4 Practical Applications

1.4.1 Airfoil Design

Aerodynamic shape optimization emerged after computational fluid dynamics (CFD) started to be used more routinely. Once aerodynamicists were able to analyze arbitrary shapes, the natural next step was to devise algorithms that improved the shape automatically [4, 5]. Initially, 2D problems where tackled, resulting in airfoil shape optimization. Today, full configurations can be optimized not only considering aerodynamics [13, 14], but aerodynamics and structures [9].

The airfoil design optimization example we show here are due to Nemec et al. [10]. The solution shown in Figure 1.4 corresponds to the maximization of the lift-to-drag ratio of the airfoil with respect to 9 shape variables and angle of attack, subject to airfoil thickness constraints.

Instead of maximizing the lift-to-drag ratio, another possible objective function is the drag coefficient. In this case, we must impose a lift constraint. Figure 1.5 shows the results for a drag minimization problem. We can see that the optimized results minimizes the drag at the specified

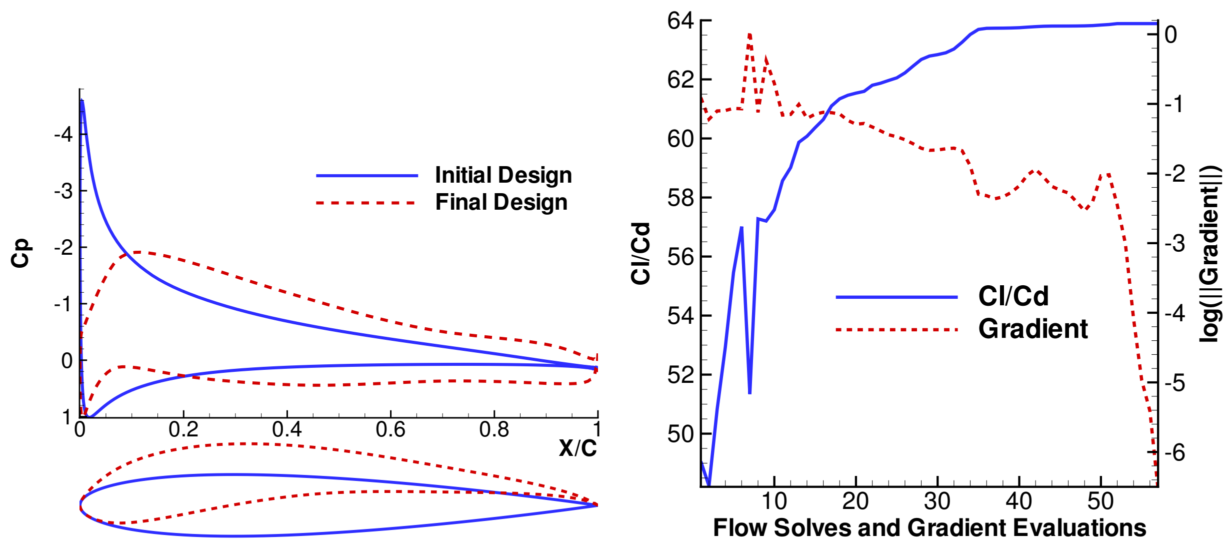


Figure 1.4: Airfoil shapes, corresponding pressure coefficient distributions, and convergence history for lift-to-drag maximization case.

condition to a fault and exhibits particularly bad off-design performance. This is a typical example of the optimizer exploiting a weakness in the problem formulation.

To improve off-design performance, we can include the performance of multiple flight conditions in the objective function. This is the case for the problem show in in Figure 1.6, in which four flight conditions where considered.

Figure 1.7 shows the solution of a multiobjective problem with drag and the inverse of lift coefficient as two objective functions. The solution is a line with an infinite number of designs called the *Pareto front*.

1.4.2 Structural Topology Optimization

This example is from recent work by James and Martins [3]. The objective of structural topology optimization is to find the shape and *topology* of a structure that has the minimum compliance (maximum stiffness) for a given loading condition.

Find the shape and *topology* of a structure that has the minimum compliance (maximum stiffness) for a given loading condition.

1.4.3 Composite Curing Cycle Optimization

This example tackles the problem of optimizing the temperature input in a composite curing cycle [6]. The objective is to maximize the tensile strength of the resulting composite beam.

1.4.4 Aircraft Design with Minimum Environmental Impact

The environmental impact of aircraft has been a popular topic in the last few years. In this example, Antoine and Kroo [1] show the trade-offs between cost, noise, and greenhouse gas emissions, by solving a number of multiobjective problems.

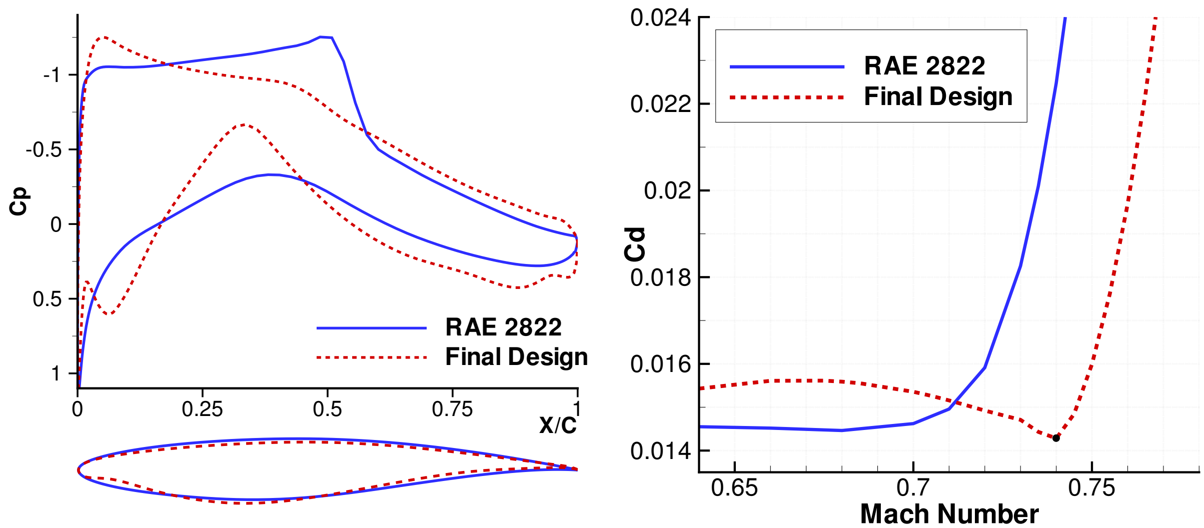


Figure 1.5: Airfoil shapes, corresponding pressure coefficient distributions, and convergence history for single point drag minimization.

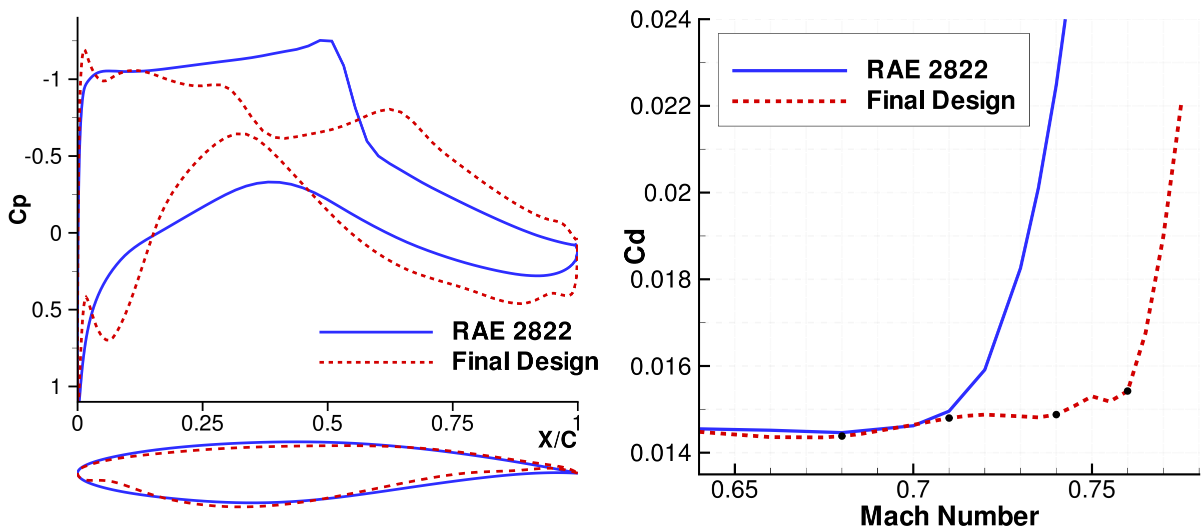


Figure 1.6: Airfoil shapes, corresponding pressure coefficient distributions, and convergence history for multipoint drag minimization.

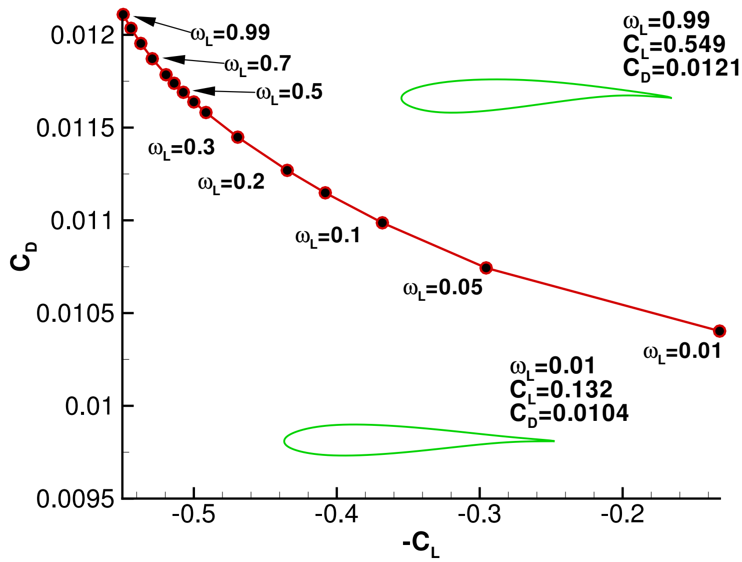


Figure 1.7: Pareto front for drag minimization and lift maximization problem

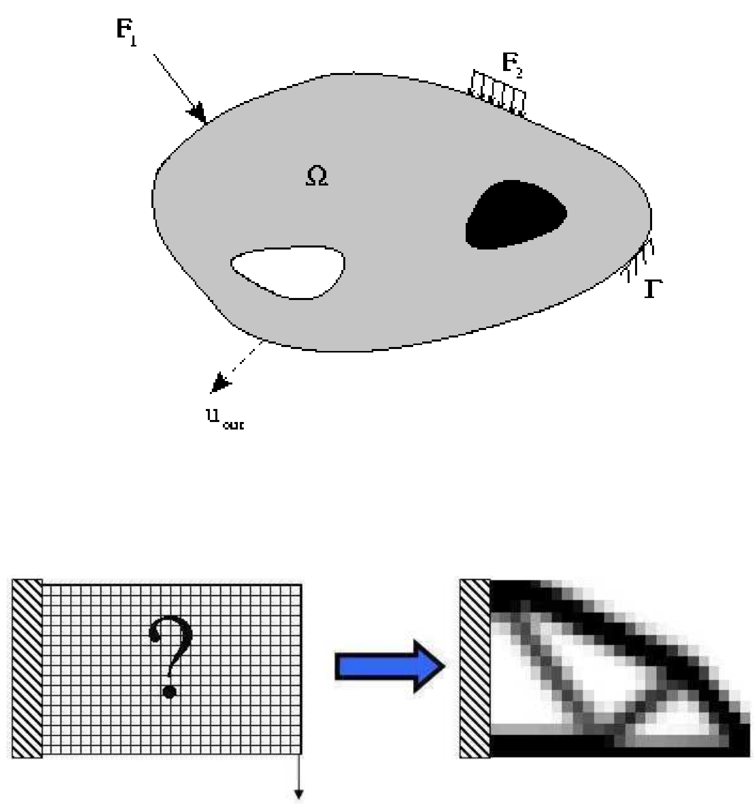


Figure 1.8: Structural topology optimization process

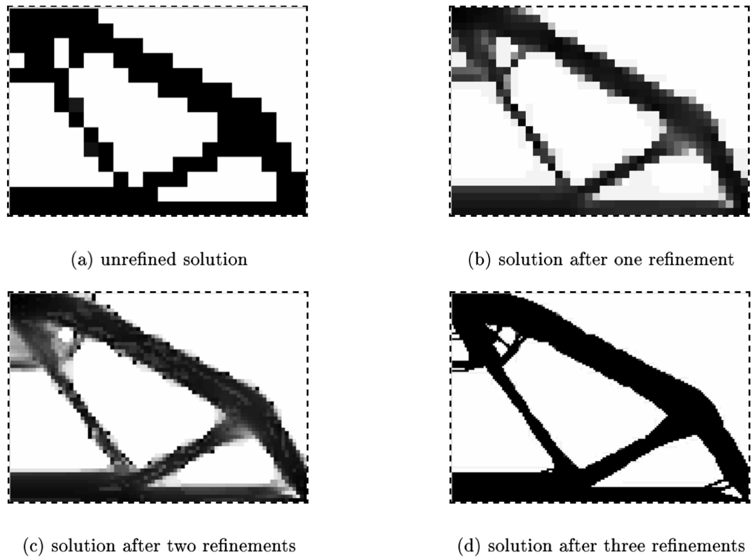


Figure 1.9: Optimized topology showing four different levels of refinement

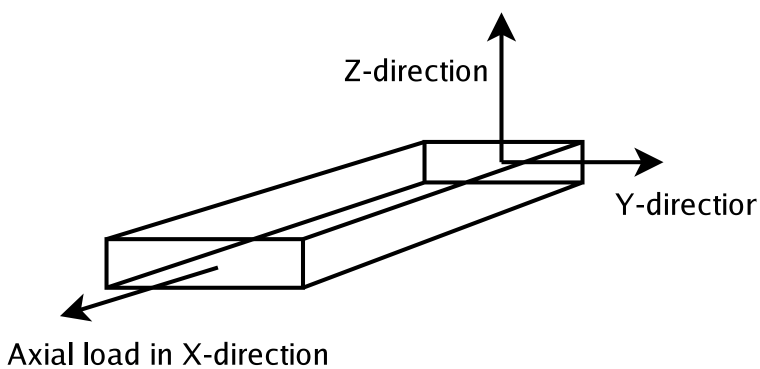


Figure 1.10: Composite beam

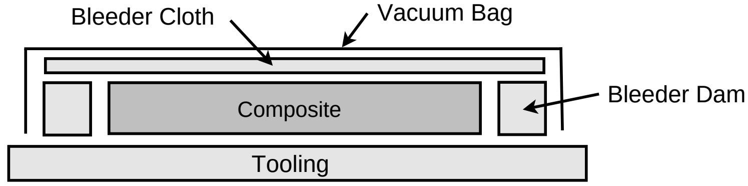


Figure 1.11: Composite beam curing arrangement

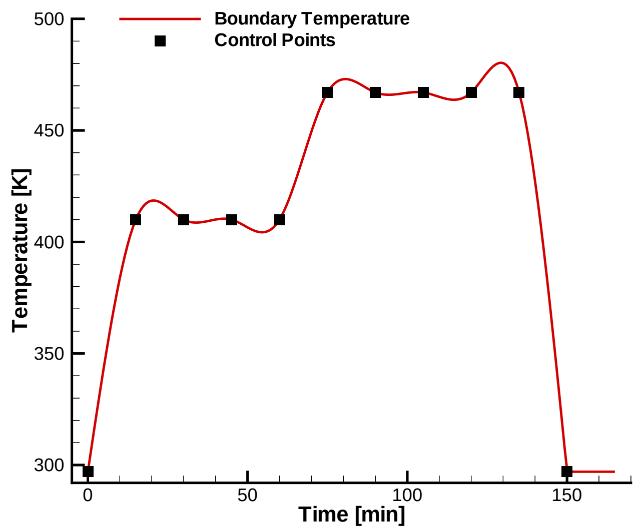


Figure 1.12: Baseline curing cycle

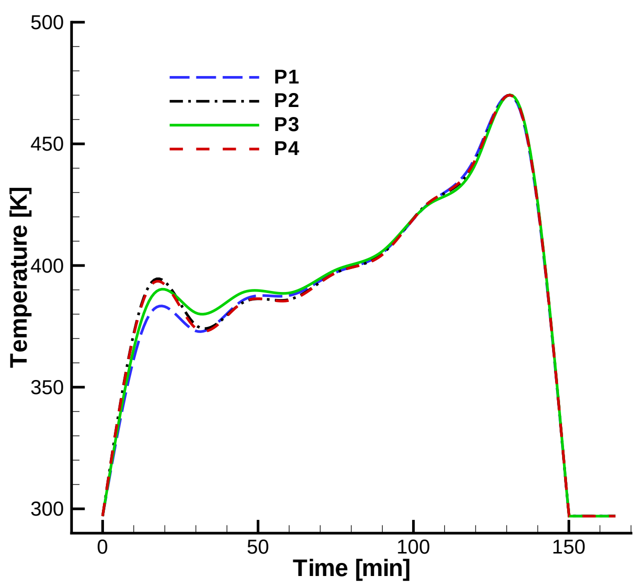


Figure 1.13: Optimized curing cycle

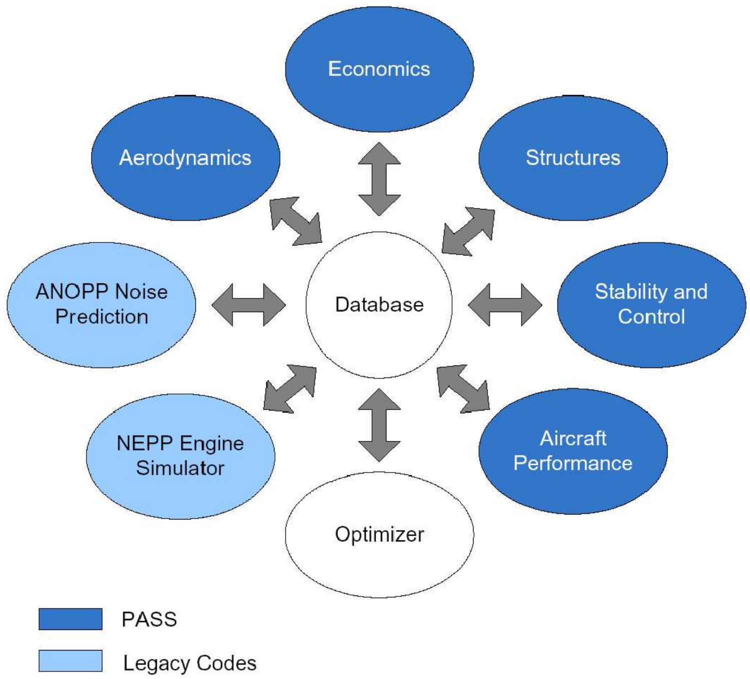


Figure 1.14: Analysis and optimization framework

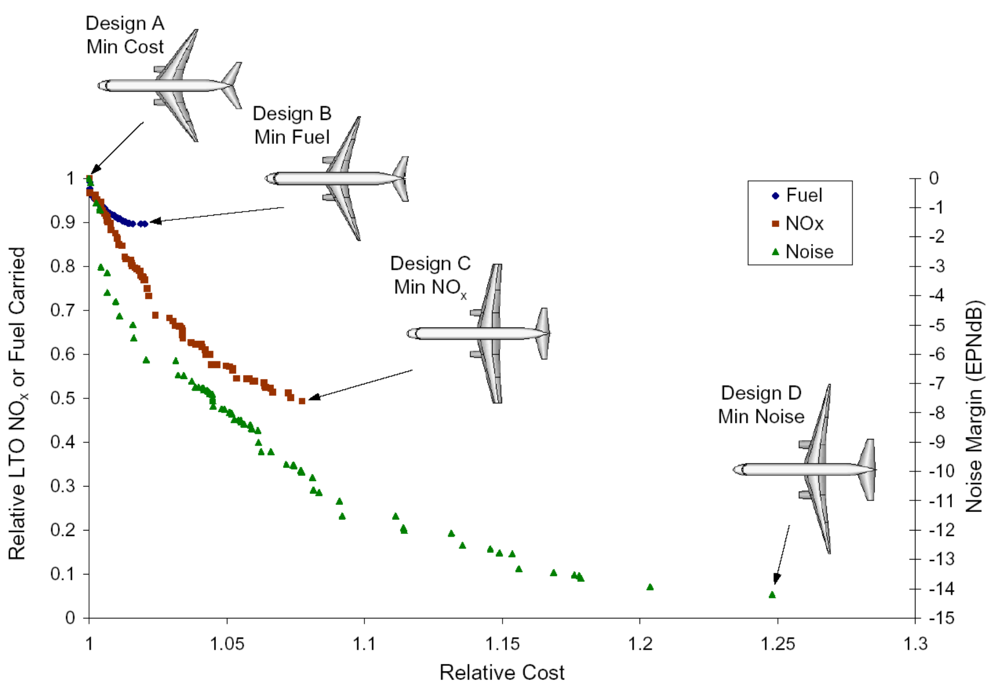


Figure 1.15: Pareto fronts of fuel carried, emissions and noise vs. operating cost

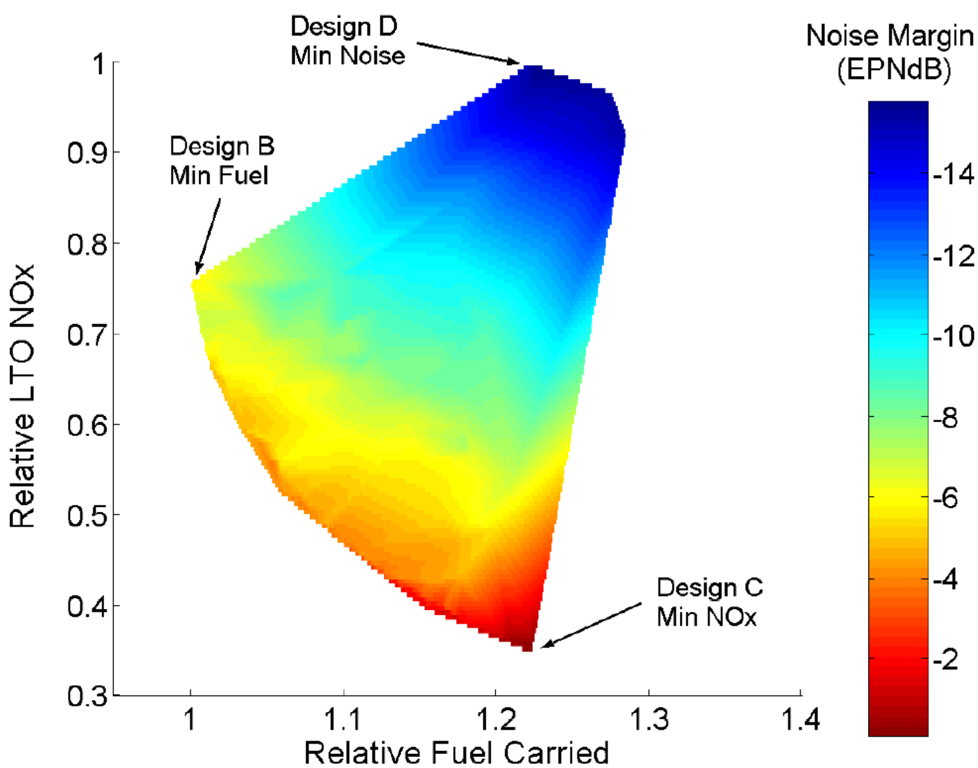


Figure 1.16: Pareto surface of emissions vs. fuel carried vs. noise

Figure 1.17: Supersonic business jet configuration

1.4.5 Aerodynamic Design of a Natural Laminar Flow Supersonic Business Jet

The goal of this project was to aid the design of a natural laminar flow wing for a supersonic business jet, a concept that is being developed by Aerion Corporation and Desktop Aeronautics [8].

In this work, a CFD Euler code was combined with a boundary-layer solver to compute the flow on a wing-body. The fuselage spoils the laminar flow that can normally be maintained on a thin, low sweep wing in supersonic flow. The goal is to reshape the fuselage at the wing-body junction to maximize the extent of laminar flow on the wing. Three design variables were used initially, with quadratic response surfaces and a trust region update algorithm.

The baseline design, whose solution is shown in Figure 1.19, is a Sears–Haack body with wing. This results in early transition (the white areas in the boundary-layer solution). N^* is the measure of laminar instability, with 1.0 (white) being the prediction of transition. The flow is then turbulent from the first occurrence of $N^* = 1$ to the trailing edge irrespective of further values of N^*.

With only three design variables (the crosses on the fuselage outline that sit on the wing) and two iterations (not even near a converged optimization) the improvement is dramatic.

With five design variables, and a few more trust-region update cycles, a better solution is found.

The boundary layer is much farther from transition to turbulent flow as can be seen by comparing the green and yellow colors on this wing with the red and violet colors in the three variables case. Also notice how subtle the reshaping of the fuselage is. This is typical of aerodynamic shape optimization: small changes in shape have a big impact in the design.

1.4.6 Aerostructural Design of a Supersonic Business Jet

See Martins et al. [9] for details.

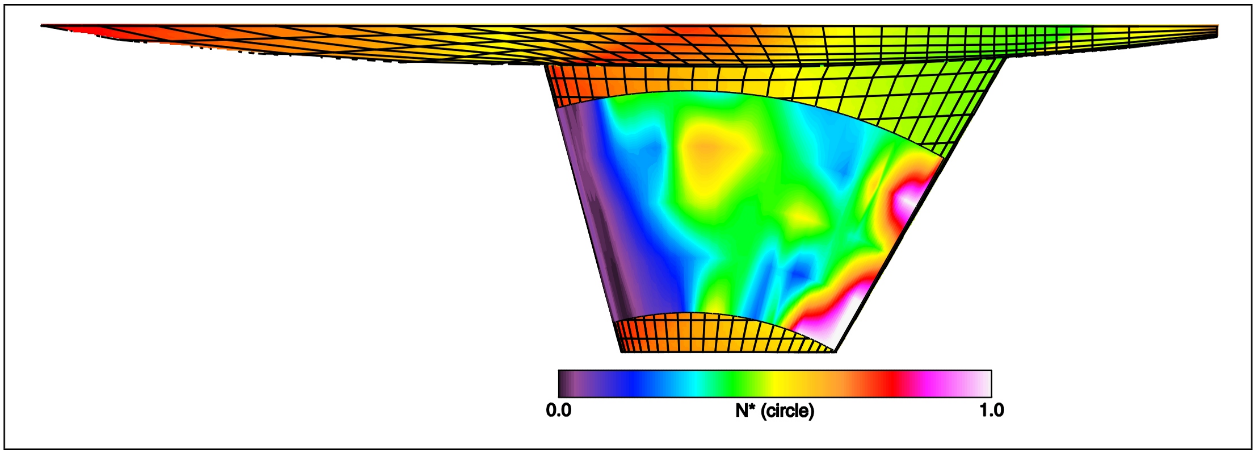


Figure 1.18: Aerodynamic analysis showing the boundary-layer solution superimposed on the Euler pressures

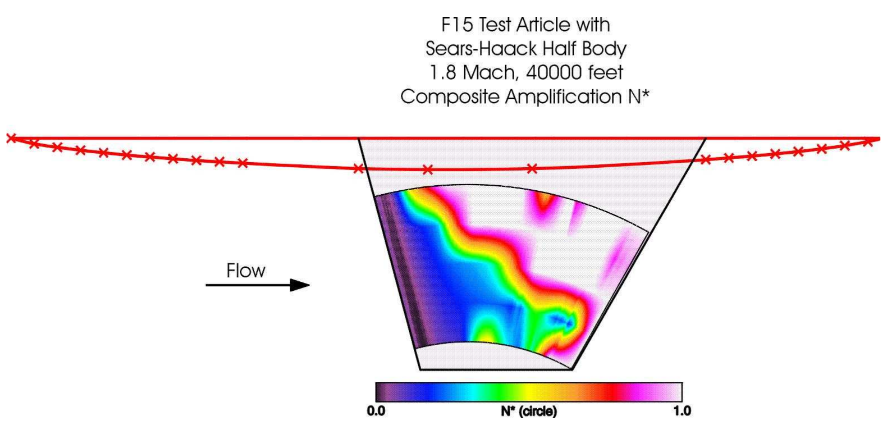


Figure 1.19: Baseline design. The colors on the wing show a measure of laminar instability: White areas represent transition to turbulent boundary layer.

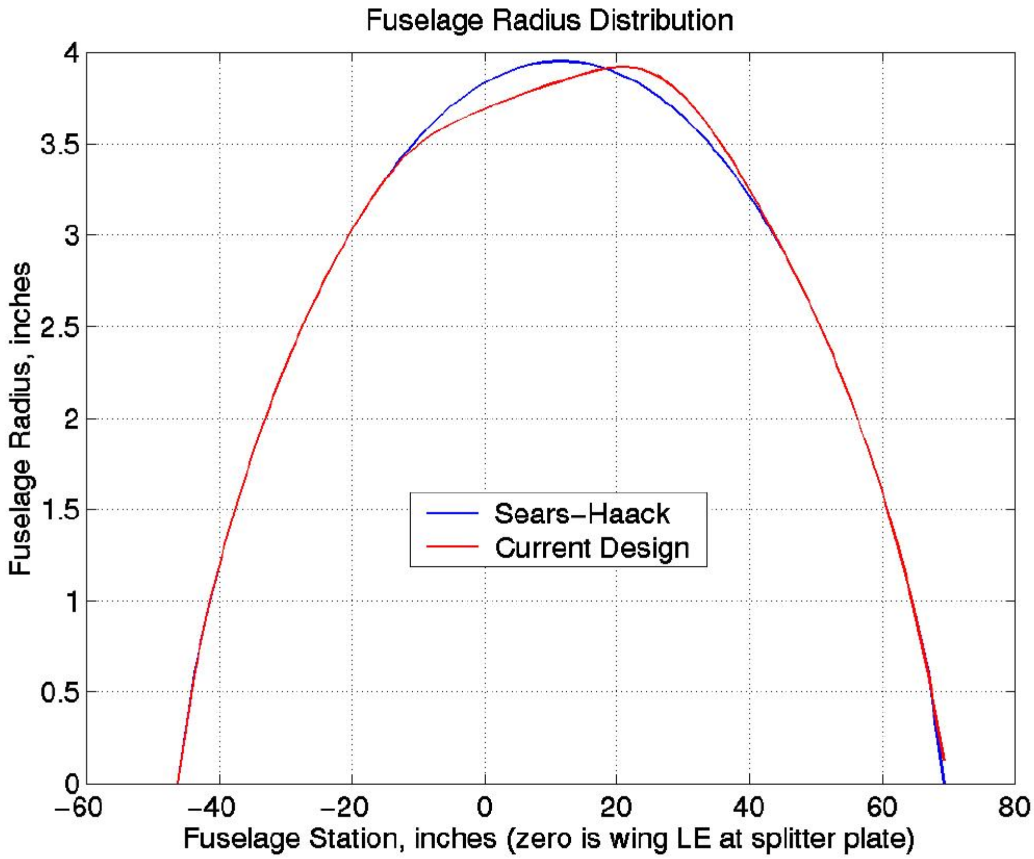


Figure 1.20: Fuselage shape optimization with three shape variables. From the nose at left, to the tail at right, this is the radius of the original (blue) and re-designed (red) fuselage after two iterations.

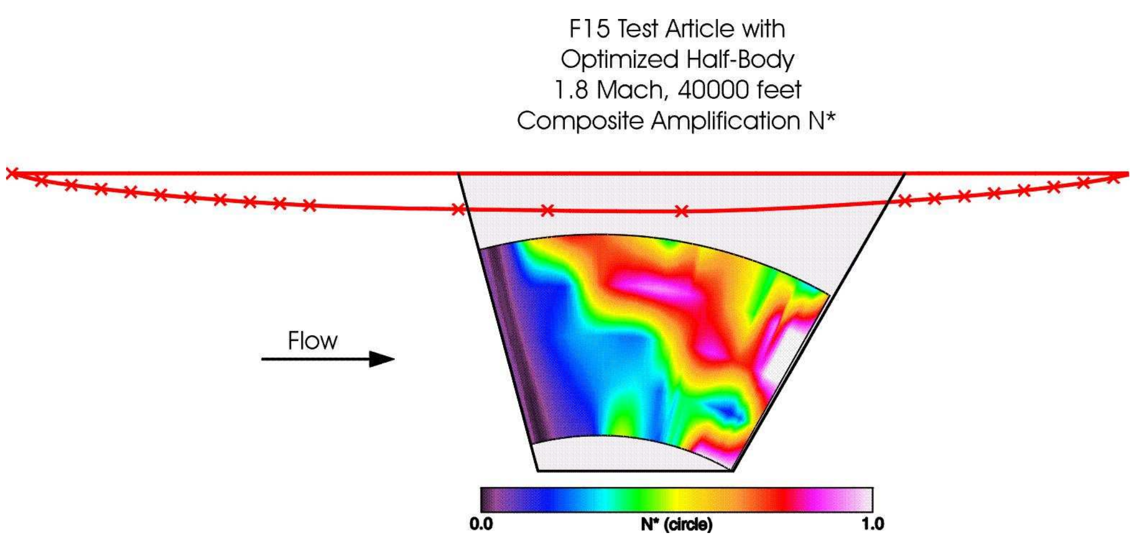


Figure 1.21: Laminar instability for fuselage optimized with three design variables

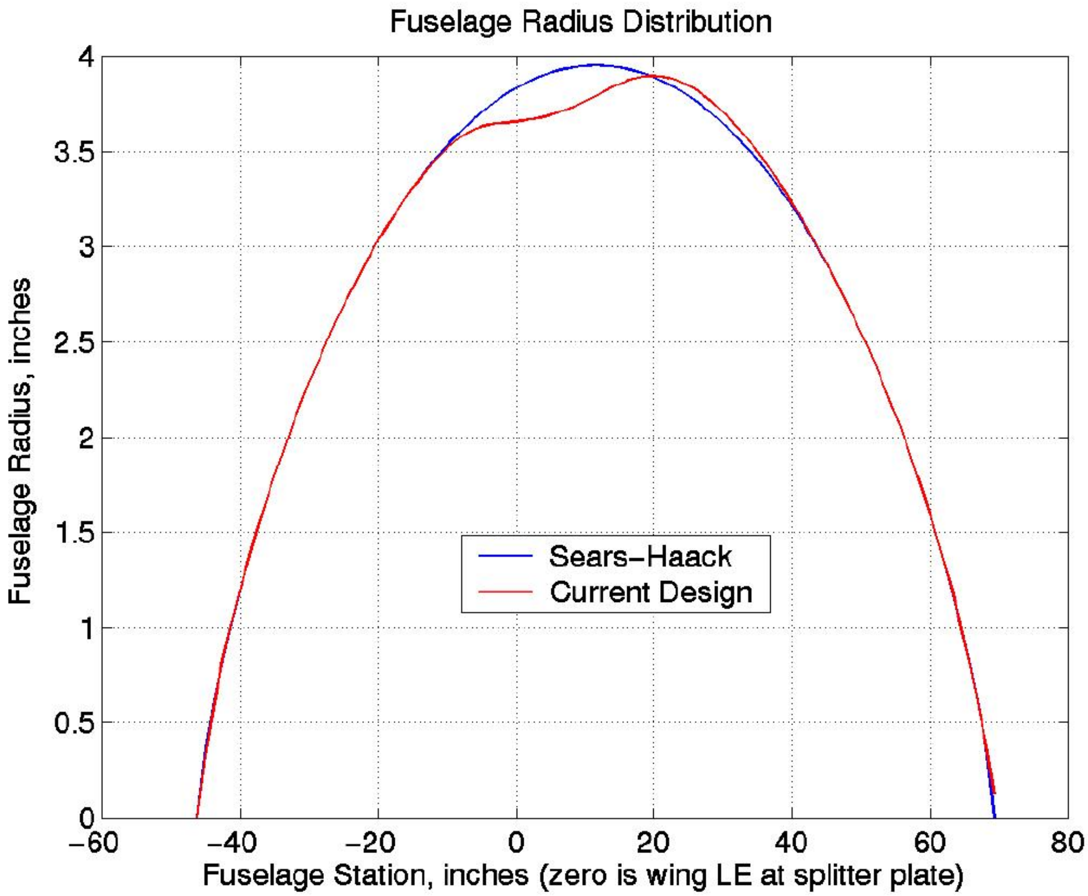


Figure 1.22: Fuselage shape optimization with five shape variables.

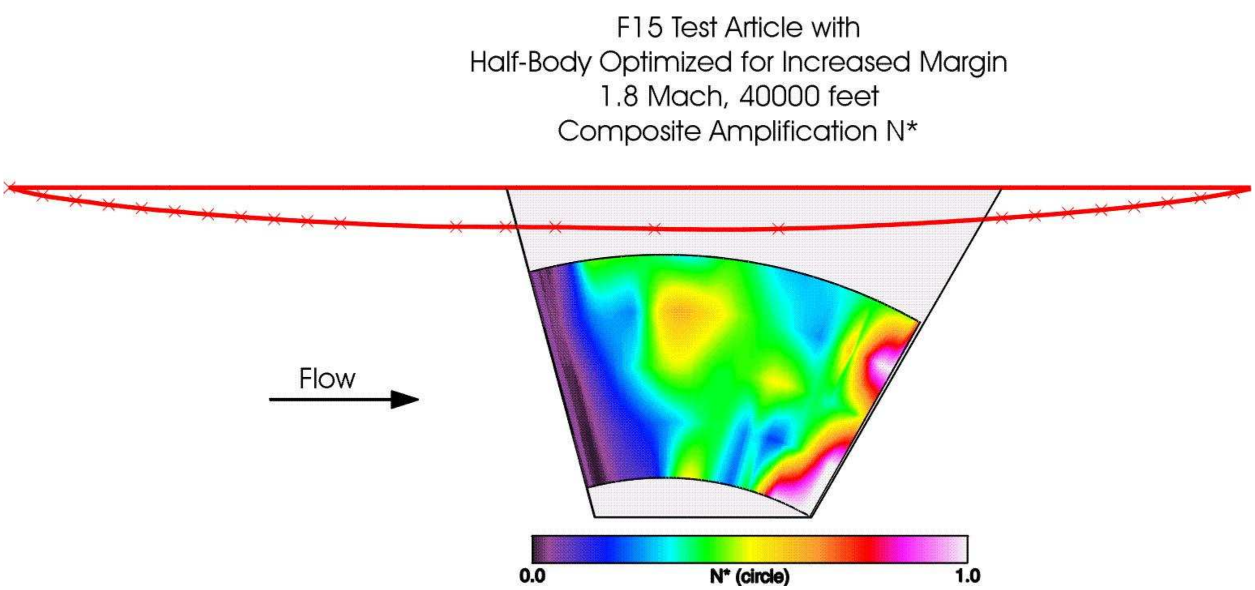
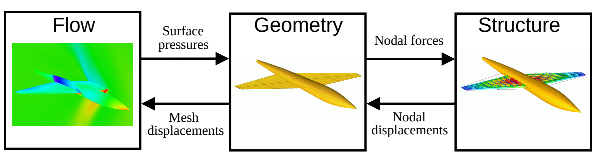
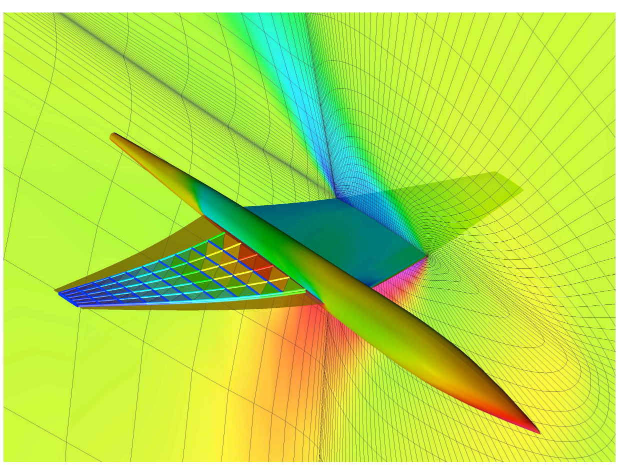


Figure 1.23: Laminar instability for fuselage optimized with 5 design variables





- Aerodynamics: a parallel, multiblock Euler flow solver.
- Structures: detailed finite element model with plates and trusses.
- Coupling: high-fidelity, consistent and conservative.
- Geometry: centralized database for exchanges (jig shape, pressure distributions, displacements)
- Coupled-adjoint sensitivity analysis: aerodynamic and structural design variables.

Supersonic business jet specification:

- Natural laminar flow

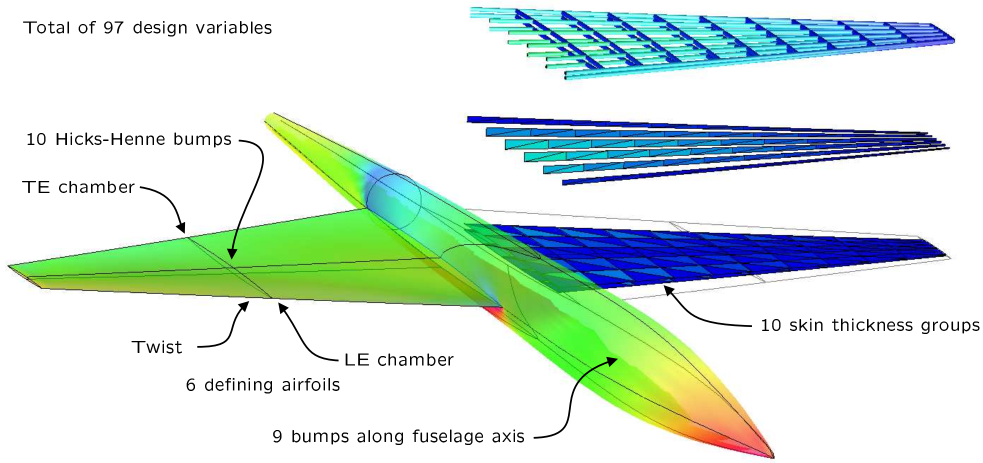


Figure 1.24: Baseline configuration and design variables

- Cruise at Mach = 1.5
- Range = 5,300nm
- 1 count of drag = 310 lbs of weight

Objective function to be minimized:

$$I = \alpha C_D + \beta W \tag{1.2}$$

where C_D is that of the cruise condition. The structural constraint is an aggregation of the stress constraints at a maneuver condition, i.e.,

$$\mathrm{KS}(\sigma_m) \geq 0 \tag{1.3}$$

The design variables are external shape and internal structural sizes.

1.4.7 Aerostructural Shape Optimization of Wind Turbine Blades Considering Site-Specific Winds

See Kenway and Martins [7] for details.

- What effect does a location's particular wind distribution have on optimum design?
- Two sites are chosen to represent the two differing environments:
- Direct simulation of the total mechanical output

Objective: Minimize the cost of energy, $\mathrm{COE} = \frac{C}{\mathrm{AEP}}$
Design Variables:

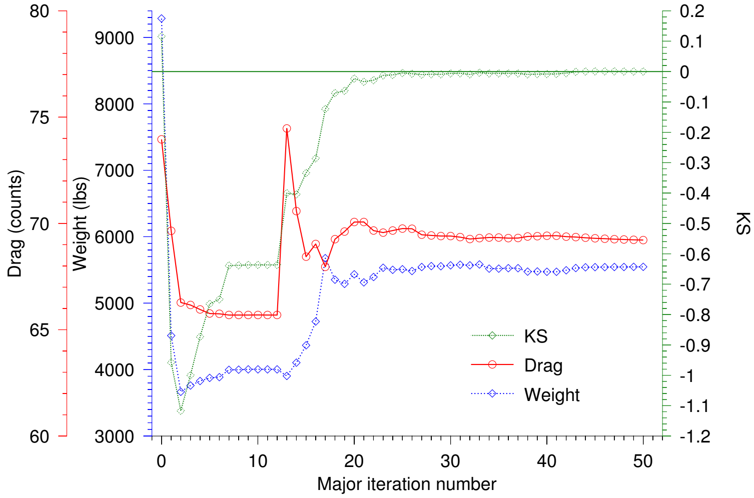


Figure 1.25: Optimization history

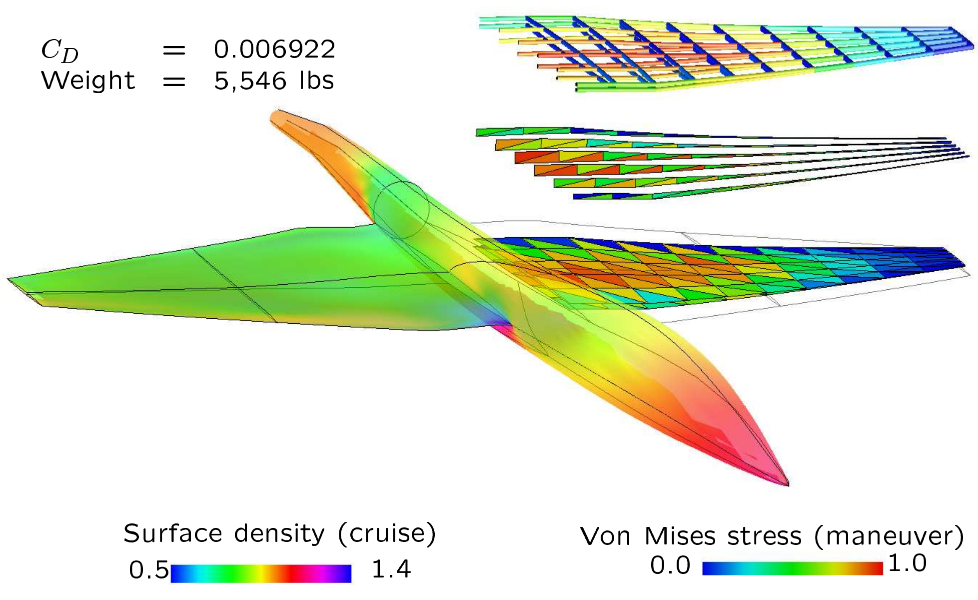


Figure 1.26: Optimized design

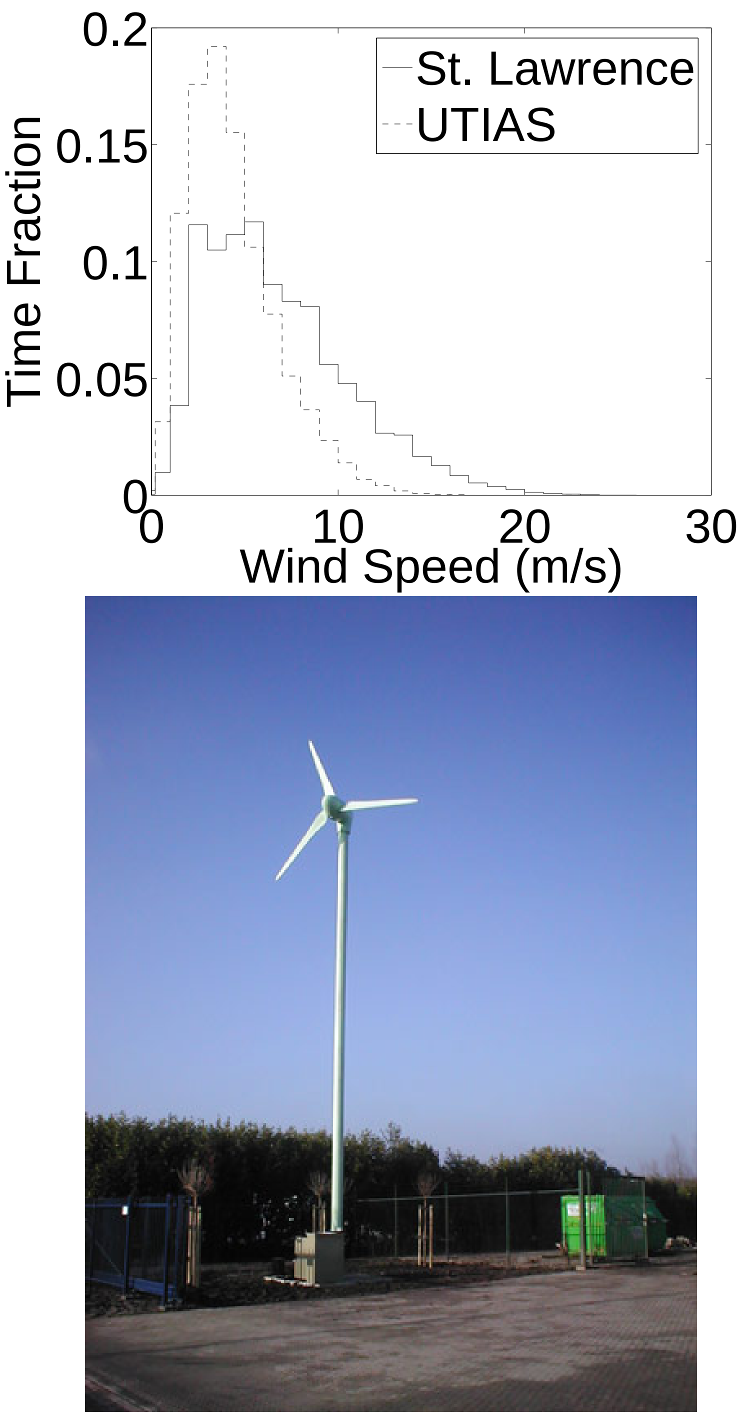
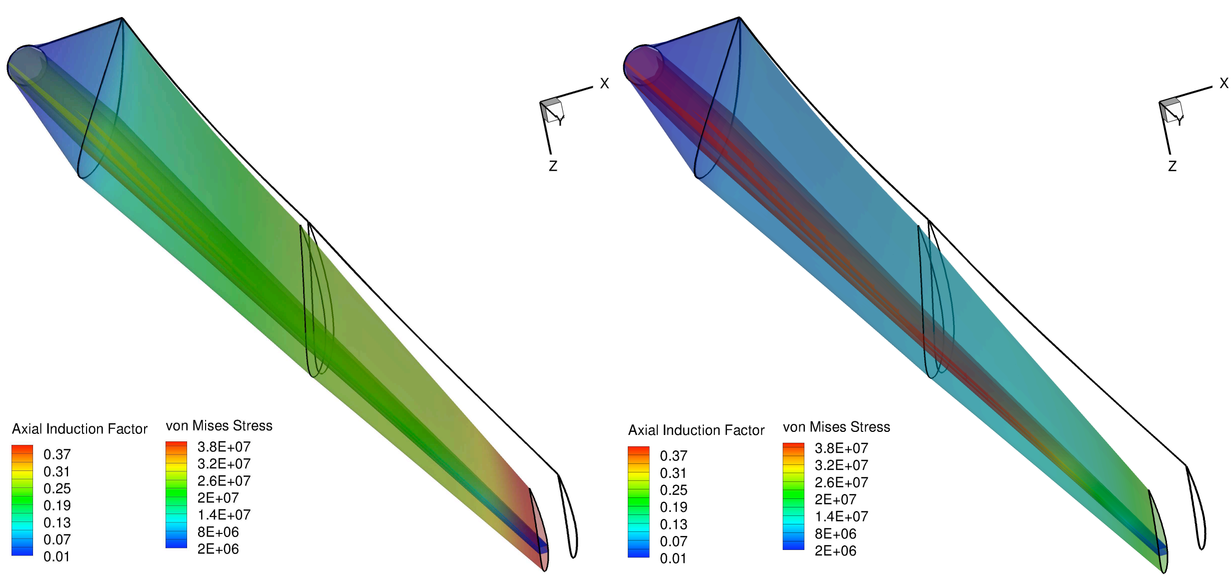


Figure 1.27: Wind speed distributions (left) and the Wes5 Tulipo wind turbine (right).

Design Variable	Count	Lower Limit	Upper Limit
Chord	4	.05 m	.40 m
Twists	4	-75 deg	75 deg
W_{spar}	4	4%	30%
t_{spar}	4	0.3 mm	10mm
t_{foil}	3	6%	20%
Ω	varies (12)	7.5 rad/s	14.7 rad/s

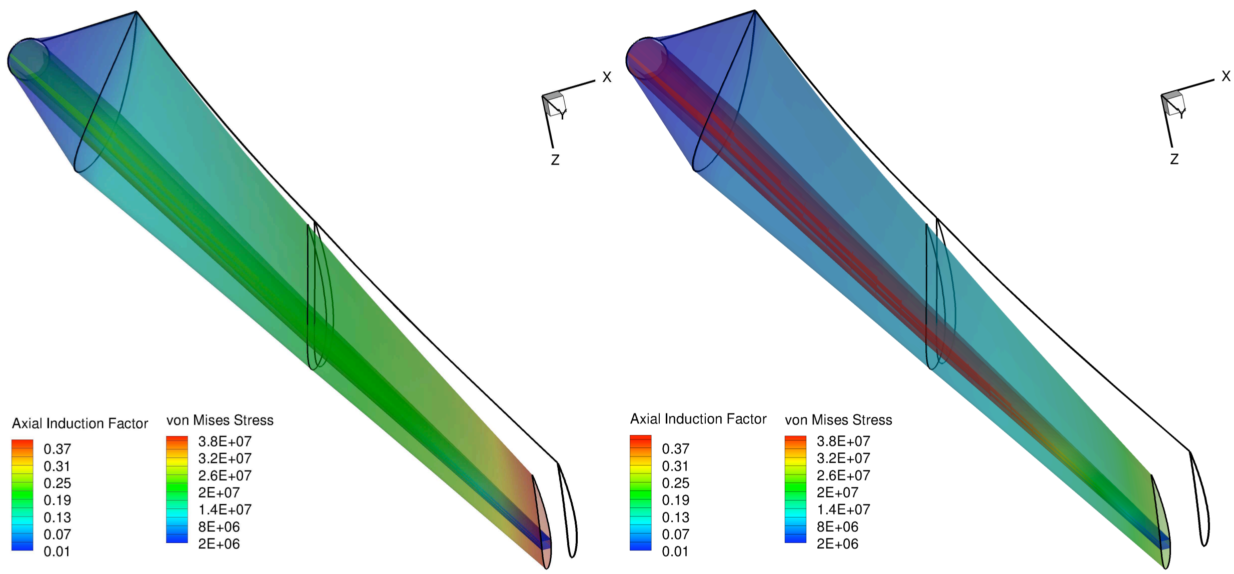
Constraints:

Constraint	Minimum	Maximum
Stress	-	40MPa
Spar Mass	-	3.7kg
Surface Area	-	0.83m^2
Power	-	5000 W
Geometry	0.5mm	-



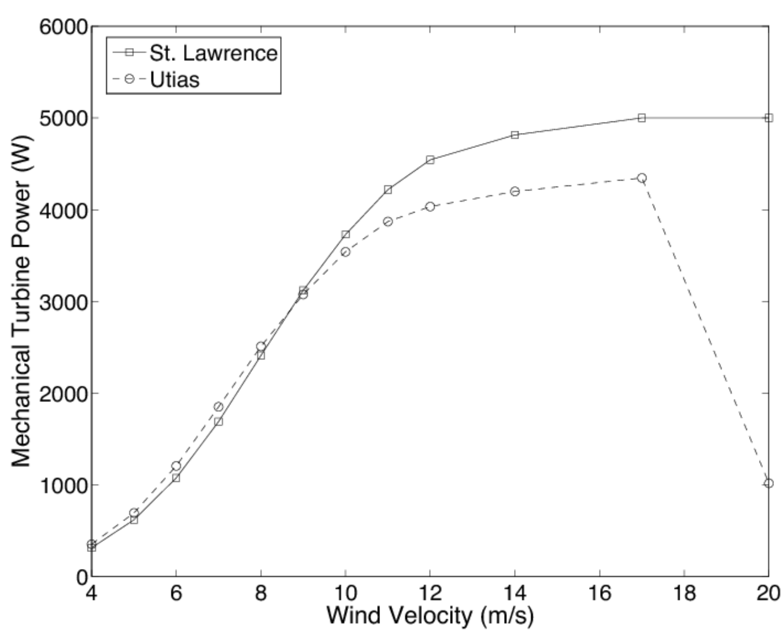
7m/s 12m/s

Location	$\bar{P}_{\text{init}}$ (W)	$\bar{P}_{\text{opt}}$ (W)	$\bar{P}_{\text{other-opt}}$ (W)	Site-specific increase
St. Lawrence	1566.1	1984.5	1905.1	4.17%
UTIAS	660.2	853.3	826.0	3.31%



7 m/s

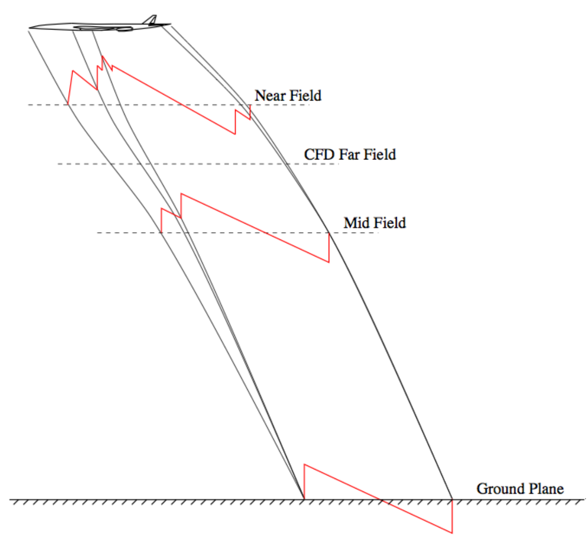
12 m/s



1.4.8 MDO of Supersonic Low-boom Designs

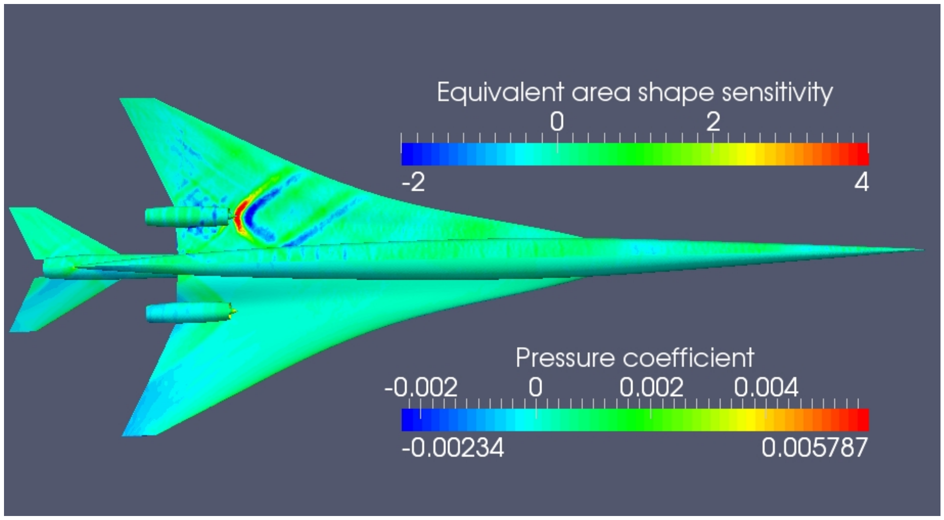
The goal in low-boom design is to reduce the strength of the sonic boom sufficiently to permit supersonic flight overland. The idea is to make subtle changes to the shape of the aircraft that lead to changes in the sonic boom at ground level. This is a very challenging MDO problem. For example, the shape has a direct impact on both the aerodynamic performance and the sonic-boom. In addition, the strength of the sonic boom is related to the weight of the aircraft and length of the aircraft.

One idea for low-boom design is to find (somehow) a target equivalent-area distribution that produces an acceptable ground level boom. Subsequently, we can use CFD-based inverse design to shape an aircraft to match the desired target equivalent area. This is the approach taken in Palacios [11].



Propagation of the sonic boom from the aircraft near field to the ground.

Early geometry for the Nasa/Lockheed N+2 configuration.



Pressure field (lower half) and equivalent-area shape sensitivity fields.

Bibliography

[1] Nicolas E. Antoine and Ilan M. Kroo. Aircraft optimization for minimal environmental impact. *Journal of Aircraft*, 41(4):790–797, 2004.

[2] Raphael T. Haftka. Optimization of flexible wing structures subject to strength and induced drag constraints. *AIAA Journal*, 14(8):1106–1977, 1977.

[3] Kai James, Jorn S. Hansen, and Joaquim R. R. A. Martins. Structural topology optimization for multiple load cases using a dynamic aggregation technique. *Engineering Optimization*, 41 (12):1103–1118, December 2009. doi: 10.1080/03052150902926827.

[4] A. Jameson. Aerodynamic design via control theory. *Journal of Scientific Computing*, 3(3): 233–260, sep 1988.

[5] Antony Jameson. Computational aerodynamics for aircraft design. *Science*, 245:361–371, 1989.

[6] Graeme Kennedy. Manufacturing process optimization for improved failure performance of thick composite structures. Master's thesis, University of Toronto Institute for Aerospace Studies, 2007.

[7] Gaetan Kenway and Joaquim R. R. A. Martins. Aerostructural shape optimization of wind turbine blades considering site-specific winds. In *Proceedings of the 12th AIAA/ISSMO Multidisciplinary Analysis and Optimization Conference*, Victoria, BC, September 2008. AIAA 2008-6025.

[8] Ilan M. Kroo, Richard Tracy, James Chase, and Peter Sturdza. Natural laminar flow for quiet and efficient supersonic aircraft. *AIAA Paper* 2002-0146, 2002.

[9] Joaquim R. R. A. Martins, Juan J. Alonso, and James J. Reuther. High-fidelity aerostructural design optimization of a supersonic business jet. *Journal of Aircraft*, 41(3):523–530, 2004. doi: 10.2514/1.11478.

[10] Marian Nemec, David W. Zingg, and Thomas H. Pulliam. Multi-point and multi-objective aerodynamic shape optimization. *AIAA Paper* 2002-5548, 2002.

[11] Francisco Palacios, Juan Alonso, Michael Colonno, Jason Hicken, and Trent Lukaczyk. Adjoint-based method for supersonic aircraft design using equivalent area distributions. In *50th AIAA Aerospace Sciences Meeting*, number AIAA-2012-0269, Nashville, Tennessee, United States, 2012.

[12] O. Pironneau. On optimum design in fluid mechanics. *Journal of Fluid Mechanics*, 64:97–110, 1974.

[13] James J. Reuther, Antony Jameson, Juan J. Alonso, , Mark J. Rimlinger, and David Saunders. Constrained multipoint aerodynamic shape optimization using an adjoint formulation and parallel computers, part 1. *Journal of Aircraft*, 36(1):51–60, 1999.

[14] James J. Reuther, Antony Jameson, Juan J. Alonso, , Mark J. Rimlinger, and David Saunders. Constrained multipoint aerodynamic shape optimization using an adjoint formulation and parallel computers, part 2. *Journal of Aircraft*, 36(1):61–74, 1999.