Massachusetts Institute of Technology
Department of Aeronautics and Astronautics

PhD Proposal Document

Advancing the Usability, Interpretability, and Practicality of Conceptual Aircraft Design Optimization with Code Transformations

Ph.D. Candidate:
Peter Sharpe

Committee:
Professor R. John Hansman (Chair), MIT
Professor Mark Drela, MIT
Professor Karen Willcox, University of Texas at Austin

External Evaluator:
Dr. Tony Tao, MIT Lincoln Laboratory

December 11, 2023

Abstract

TODO abstract

Contents

Chapter 1

Introduction

1.1 Background

TODO intro

1.2 Project Definition and Thesis Overview

1.3 Motivation for Improving Industry Access to Design Optimization

1.4 Proposal Document Organization

The latest doctoral handbook for the MIT Department of Aeronautics and Astronautics requires the following elements in a PhD proposal, which, for reader convenience, are mapped to specific portions of this document:

Table 1.1: Requirement-to-section mapping for the PhD proposal document.

Requirement	Section
A clear, specific statement of the technical problem and the objectives of the proposed research	Section 1.2, 3.1
A thorough, adequately referenced, summary of previous work done on the problem	Section 2
A plan for the initial approach to the problem	Section 3.2
An outline of the major foreseeable steps to a solution of the problem	Section 3.2, 4.4
An estimate of the time that might be required	Section 4.4
A list of the facilities needed	Section 4.5

Chapter 2

A Review of Aircraft Design Optimization

2.1 Early Promises, Predictions, and Pitfalls

> *"When an aircraft designer hears that a new program will use multidisciplinary optimization, the reaction is often less than enthusiastic. Over the past 30 years, aircraft optimization at the conceptual and preliminary design levels has often yielded results that were either not believable, or might have been obtained more simply using methods familiar to the engineers. Even 5-20 years ago, actual industry application of numerical optimization for aircraft preliminary design was not widespread."*
>
> -Ilan Kroo, 1997 [1]

These are the opening lines of a landmark 1997 paper that reviewed the state of the art and future directions in the then-nascent field of aircraft multidisciplinary design optimization (MDO) [1]. The paper, by aircraft designer and MDO pioneer Ilan Kroo, not only reviewed the status of the field's academic research, but also took the key step of assessing whether these research advances had translated to practical industry impact. As a later review by an MDO technical organizing committee would emphasize, "the ultimate benchmark of a research field's impact is indicated by the realization of its theories into successful products throughout industry." [2]

Kroo's assessment of the field is generally optimistic, but with some important reservations. On one hand, he notes the auspicious progress of aircraft MDO research during the preceding decades by all traditional metrics: problem size, analysis fidelity, runtime speed, and so on. He credits these successes largely to both algorithmic advances and the exponential growth of computational power over time (Moore's law[1]). Extrapolating these trends forward, he concludes that the field is poised to make a significant impact on the aircraft design process across both academia and industry. The promise of MDO is made clear in a remark that many aircraft designers would agree with today: "In a very real sense, preliminary design is MDO." [1]

On the other hand, Kroo notes that actual industry applications of aircraft MDO remained conspicuously limited as of the paper's 1997 publication, and "many difficulties remain in the routine application of MDO." [1] Other works from the early days of aircraft MDO corroborate this dearth of industry adoption. For example, in a 1982 AIAA lecture titled *On Making Things the Best – Aeronautical Uses of Optimization*, optimization advocate Holt Ashley notes the "keen disappointment felt by many [optimization] specialists because their theories have received so little practical application." [3] Ashley goes further by conducting an exhaustive industry-wide survey to identify "successful practical applications" of aircraft design optimization. The survey received "overwhelming" industry interest and encouragement; however, "the yield of examples which met the criterion of having been incorporated into a vehicle that actually operates in the Earth's atmosphere was painfully, perhaps shockingly small."

Perhaps the reason for Ashley's pain is that, even at the time, aircraft designers in industry widely recognized the immense potential of MDO tools to fundamentally transform engineering design workflows. As two Lockheed engineers stated in 1998 [4], "The technical community knows the power of MDO and not having a cradle-to-grave example has been a continual source of frustration, as voiced by AIAA MDO technical committee members [for] years." Similarly, a 2002 Boeing paper identifies MDO as one of four key technologies poised to define the next generation of aircraft design[2] [5].

[1] "Moore's law" is an empirical observation that chip transistor density (a surrogate for computational power) has tended to double roughly every two years, a trend that has held true for the past half-century.

[2] With the others being computational simulation, small uncrewed aircraft, and newly-emphasized design considerations such as environmental footprint and operations optimization.

Motivated by this gap between promise and practice, Kroo and other luminaries offer several reasons for the lack of industry adoption of MDO technologies:

1. **Inadvertently-violated model assumptions:** When optimization is applied to an analysis toolchain, it acts adversarially, disproportionately seeking out the "weakest link" in the analysis chain and exploiting it. Simplified models that are acceptable in a manual sizing study are often unacceptable in an optimization study, because implied assumptions that an engineer would naturally cross-check during manual design are prone to optimizer exploitation. This can lead to unrealistic results from MDO tools that degrade user trust in optimization processes. Kroo contends that the solution to this problem is to implement higher-fidelity models that account for more edge cases, an approach we explore later in Section 2.3.1.

2. **Missing models and constraints:** Critical aircraft design choices are often determined by tradeoffs spanning multiple disciplines. If an MDO tool does not incorporate relevant disciplines (e.g., it performs aerodynamic, propulsive, and structural analyses, but the true design driver is noise), then the resulting design will be fundamentally flawed with no indication to the user whatsoever. This can lead to costly design mistakes. Indeed, Kroo suggests that a flawed MDO tool is not only useless, but worse, due to the false confidence it can instill.

3. **Challenges of managing complexity:** As computational power grows, MDO tools can incorporate increasingly numerous and higher-fidelity models. To first order, when the number of models N grows, the potential number of cross-discipline couplings to manage tends to scale as $\mathcal{O}(N^2)$. Therefore, in the limit of growing computational power, the practical bottleneck is less about implementing individual analyses, but more about managing this communication overhead and architecting the optimization code itself.

Kroo primarily attributes these early barriers to practical industry adoption to the limited computational power of the era, noting: "This convergence between computational capability and computational requirements for interesting design problems is one of the reasons that MDO is considered to be such a promising technology, despite the limited acceptance of pioneering

MDO efforts.” A contemporaneous review by Sobieski and Haftka agreed, identifying “very high computational demands” as a “major [obstacle] to realizing the full potential of MDO”. [6]

However, some works recognized that not all challenges would recede with increasing computational power. A 2002 review paper by a Boeing Technical Fellow in *Journal of Aircraft* [5] cautions: “New [MDO] strategies need to be developed. . . which take advantage of the assumptions and techniques that airplane designers use, rather than letting a computer churn away and come up with theoretically possible, but practically impossible, configurations.” Likewise, Kroo, Sobeiski, and Haftka all cite “managing complexity” as a growing challenge of MDO [1, 6]. Drela’s aptly-titled 1998 work *Pros & Cons of Airfoil Optimization* also demonstrates a similar issue [7]. Here, Drela demonstrates that even seemingly-simple design optimization problems will only yield practical results if the problem formulation, assumptions, and results are all precisely understood by an expert designer. These works presciently foreshadow modern concerns around MDO interpretability and other user frictions, urging the development of human-centered MDO approaches that synthesize mathematical optimization with designer intuition.

In summary, while computational limits were a clear early barrier, foundational challenges around managing complexity and aligning optimization with real-world design constraints were already emerging. These issues would become increasingly central as MDO research rapidly expanded in scope.

2.2 A Retrospective on Aircraft Design Optimization in Industry

Current Status

With the benefit of an additional quarter-century of hindsight since the date of these early MDO assessments, we can begin to assess how these forecasts have played out. In many ways, these predictions by Kroo and others were remarkably accurate. The scale and speed of optimization problems solved today has indeed grown exponentially in the years since. This is not only due to increasing computational power, but also from algorithmic and architectural improvements in MDO and optimization more broadly[3]. Numerous high-quality aircraft MDO case studies and post-hoc design studies have been published in the years since, including the D8 transport aircraft [8, 9], the STARC-ABL aircraft [10], and the Aerion AS2 [11]. Some of these studies leverage relatively-high-fidelity models (e.g., RANS CFD) that would have been computationally-intractable for optimization studies in prior eras.

However, many of the barriers to practical industry use Kroo identified have not disappeared; to the contrary, as computational power increases, the challenges of interpretability and managing complexity ring even truer today. As a result, this gap between academic research and practical industry adoption has not closed, and in some ways is wider than ever before. A 2010 review paper [2] concludes that "the actual use of genuine MDO methods within industry at large. . . is still rather limited." In 2013, Hoburg and Abbeel lamented that "despite remarkable progress in MDO, the complexity and diversity of modern aerospace design tools and teams makes fully coordinated system-level optimization a monumental undertaking." [12] As recently as 2017, a team of Airbus engineers and MDO researchers concurred: "While the field of MDO techniques has tremendously grown since [the 1980s] in the scientific community, its application in industry is still often limited[. . .] A major challenge remains to apply MDO techniques to industrial design processes." [13]

Despite the prevalence of on-paper design studies using MDO, it remains somewhat rare to see *built-and-flown* airplanes developed with documented, significant use of MDO methodologies. Indeed, the industry norm for aircraft conceptual design remains largely unchanged: expert-driven manual design

[3] discussed later in Section 2.3

guided primarily by point analyses and parametric surveys. Here, the human designer informally fulfills an optimizer-like role, but the requirements are never explicitly translated into a formal mathematical optimization problem[4]. High-quality aircraft design case studies that implement this expert-driven manual approach successfully are those of the Joby Aviation S2 eVTOL [14] and Perdix micro-UAV [15] aircraft. Anecdotally, the industry conceptual designer's computational weapon of choice is still more often an Excel spreadsheet than a formalized MDO framework.

To quantify this gulf between academic and industry use of MDO, we surveyed industry literature and design reviews to identify aircraft development programs showing three minimal criteria:

1. A design optimization problem for a complete aircraft coupling at least three disciplines (e.g., aerodynamics, structures, and propulsion) that is solved computationally.
2. Evidence that the optimization result (or at least, some insight gained from it) was used to inform the design of aircraft.
3. Evidence that the aircraft was subsequently built and flown.

It is worth pausing to discuss why this restrictive *built-and-flown* aircraft requirement is used, since it accounts for the majority of the aircraft design studies that were excluded from this survey. Successful optimization on built-and-flown aircraft forces a level of completeness, trust, and durability that may not be present in a paper study. Ashley notes that a practical MDO result "must also survive the gauntlet of ground testing, reliability, demonstration, flight verification, and the like without further special attention". [3] Myriad other practical considerations could be added to this list – certifiability, manufacturability, lifecycle costs, maintainability, robust off-design performance, and the realities of engineering culture (e.g., can the MDO tool give not only a result, but sufficient evidence that it should be believed?), to name only a few. To justify the substantial capital of building and flying a new aircraft, optimization results must withstand a much higher standard of scrutiny and reviewability than they might otherwise.

Consistent with previous surveys and reviews [1–3, 6, 13], we find that relatively few programs meet these criteria; here, we make note of some that do. The Lockheed Martin F-22 Raptor program was one of the first programs to leverage an MDO-like process for a complete, flown aircraft design, as

[4] i.e., with a precisely specified objective, defined variables, and enumerated constraints

reported by Radovcich and Layton [4]. However, the workflow documented in this 1998 work differs significantly from modern MDO processes in that it was remarkably manual – a fusion of traditional and MDO-based design processes. While some disciplines (aerodynamics, structures, and control law design) are coupled computationally, many other disciplines (low-observability, manufacturability, etc.) are coupled in by querying teams of subject-matter experts at iterates within the optimization loop itself, essentially serving as a black-box function call. When contrasted with more-typical MDO processes where all interdisciplinary communication is computational, this manual approach has some pros and cons. One one hand, intermediate optimization iterates are continuously human-reviewed, potentially boosting trust. On the other hand, it also sharply increases the optimization runtime, from hours to months. Even allowing for this broader definition of MDO, however, the authors note the rarity of their experience in industry: "Documented experiences of MDO applications during the engineering, manufacturing, and design phases of fighter aircraft programs are not numerous. Documentation is even rarer for aircraft that have flown."

The Lockheed Martin X-59 Quiet Supersonic Transport (QueSST) is another program that leveraged an MDO-based design process throughout aircraft development, unifying aerodynamics, acoustics, and structural analyses for outer mold line design and composite ply scheduling [16–18]. (Although the X-59 has not yet met the "flown" criterion at the time of writing, industry reports credibly suggest this is imminent with no remaining obvious barriers.) The X-59 program builds upon several decades of successful MDO research for sonic-boom-minimization problems, a topic where computational shape optimization has proven particularly useful due to the non-intuitive and sensitive nature of the design problem [19].

The Airbus *Vahana*, a single-seat eVTOL demonstrator flown in 2018, is perhaps one of the most recent public examples of an MDO-based process used to develop a flown aircraft [20–22]. The initial sizing study considered aerodynamics, structures, propulsion, and cost analysis to drive conceptual trade studies between various vehicle configurations. Practical constraints and margins, such as reserve mission energy, battery cycle life, and engine-out safety (by enforcing either autorotation capability, motor redundancy, or a mass allocation for a ballistic parachute, depending on configuration), were included.

Several other programs have built and flown uncrewed research aircraft with MDO use, such as the Facebook *Aquila* solar-powered UAV [23], the

MIT *Jungle Hawk Owl* long-endurance UAV [24], and the X-48B blended-wing-body demonstrator [25–27]. Other programs document MDO usage for narrower subsystem- or component-level design, as in the detailed design of the Boeing 787 Dreamliner [2].

Aside from this, documented industrial use of MDO for complete-aircraft design remains relatively uncommon, especially in light of its widely-recognized potential utility. This lack of use can be partially explained as a lag effect stemming from long timelines of new aircraft development programs; it takes time for new tools to proliferate due to sunk-costs on design pipelines for existing programs. Indeed, the present literature review finds that documented industry use of MDO, while still uncommon, appears somewhat less scarce than previous surveys indicate [1–3, 6, 13]; this suggests that industry use may be gradually increasing. However, as aircraft MDO research and substantial industry interest in optimization dates back roughly forty years [3, 28], this lag effect alone does not constitute a complete explanation.

Ashley considers another possible explanation for the lack of visible MDO use in industry: that of "military classification or company proprietary considerations." [3] However, after exhaustive correspondence with dozens of aircraft design industry contacts, Ashley concludes that this limitation was only occasional and "not to an extent that would affect the principal conclusions." Even after decades, industry usage of formal mathematical optimization processes for aircraft design remains fairly limited, and, when used, often leverages only a small fraction of the field's research advances.

2.3 Pivotal Advances in Design Optimization Research

On a more optimistic note, MDO's substantial academia-industry gap is equally attributable to academia's remarkable advances in design optimization research over recent decades. This progress is readily apparent from analyzing trends in academic literature. As shown in Figure 2.1, the fraction of aircraft design publications directly referencing MDO has grown from near-zero in 1985 to 10% today.

However, this statistic alone understates the true magnitude of MDO's impact. Arguably the most pivotal contribution of MDO research to aircraft design has been to catalyze a shift towards an *optimization mindset*: the recognition that optimization is not only a useful tool, but also a principled mathematical framework that can represent complete design problems. Figure 2.1 shows that this mindset is a relatively recent development – the fraction of aircraft design publications mentioning optimization in any form has tripled since the advent of MDO, reaching a majority share today.

These optimization-based design processes have shown several benefits when used to augment traditional design methods such as point analyses, parametric surveys, and carpet plots [29]. Most obviously, formal optimization methods can lead to improved design results and allow the consideration of many more design variables. Optimization can also help human designers discover clever cross-discipline synergies that might otherwise be overlooked [7], and its rigor can reduce the likelihood of biased decisions [29]. In cases where designer intuition is exhausted, such as with unconventional configurations, optimization provides a means to identifying useful directions for further exploration [7]. Torenbeek argues that an optimization-based design process can respond more quickly to unexpected changes in program requirements, whereas manual processes often require substantial redesign effort [29]. Finally, even the exercise of translating given requirements into a quantified optimization problem can prove useful – this can help a team of designers align on goals and expose subtle discrepancies in perceived requirements[5].

Another important area where MDO research has made significant progress is in defining the relationship between the human designer and the optimizer.

[5] For example, is the objective to minimize fuel burn, or to maximize range? Is operating cost a quantity that should be strictly capped (a constraint), or instead only penalized (an objective)?

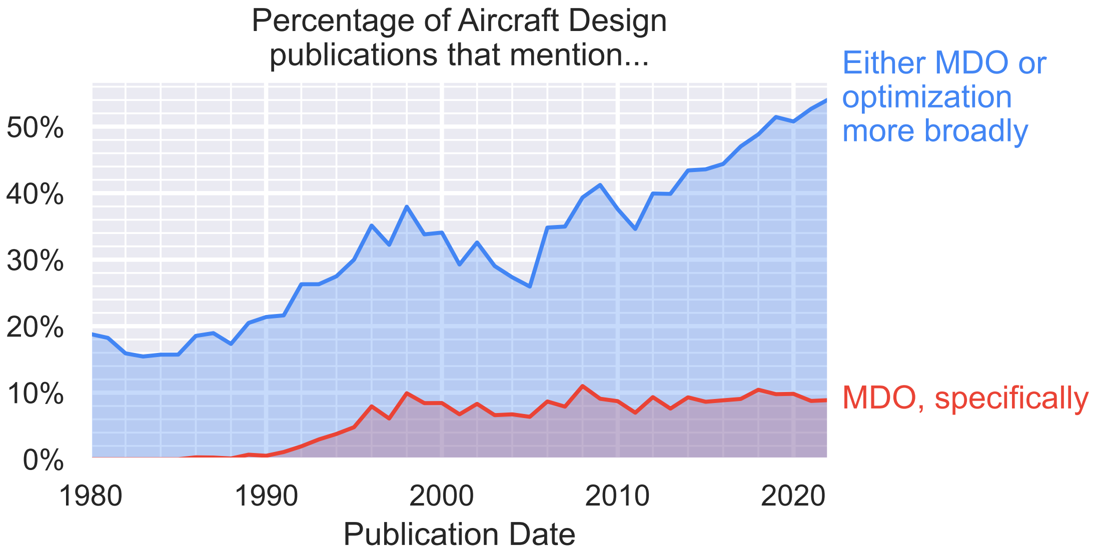

Figure 2.1: Prevalance of optimization-related keywords in academic literature with the keyword “aircraft design”. Data from Google Scholar; includes industry-standard texts such as AIAA journals and conference proceedings. MDO-specific keywords (red line) include any of: “multidisciplinary design optimization”, “MDO”, and “MDAO”. Blue line adds the keyword “optimization” to this list.

Several early aircraft design optimization works from the 1970s, with titles such as “Automating the Design Process” [28, 30, 31], speculated that optimization would eventually advance to the point that computational tools could yield complete, production-ready designs with minimal human oversight. Over time, this has largely given way to a more balanced view: although the optimization solve itself may be well-served by computational means, this forms only one small part of the larger design process. Inputting the engineer’s design intent into the optimizer accurately (“problem specification”) and extracting intuition from the results (“interpretation”) are challenges in their own right, and best addressed by iterative human-computer teams. As described by Drela in a 1998 optimization study [7], “[Engineering optimization] is still an iterative cut-and-try undertaking. But compared to [traditional] techniques, the cutting-and-trying is not on the geometry, but rather on the precise formulation of the optimization problem."

When considering this full design process (i.e., from initial requirements to product delivery), the scope of challenges becomes even broader. Indeed, industry adoption of MDO is often limited by non-computational user frictions [32–34]. In the present work, we deliberately term these user frictions "costs", borrowing optimization nomenclature to emphasize that minimizing these frictions is an implied part of the objective function of any optimization-driven design process.

Figure 2.2 summarizes a variety of design literature to present a holistic view of this complete process as applied to aircraft design [29, 32, 34–38]. Frictions that a designer may experience from a computational optimization framework at each step are shown in red.

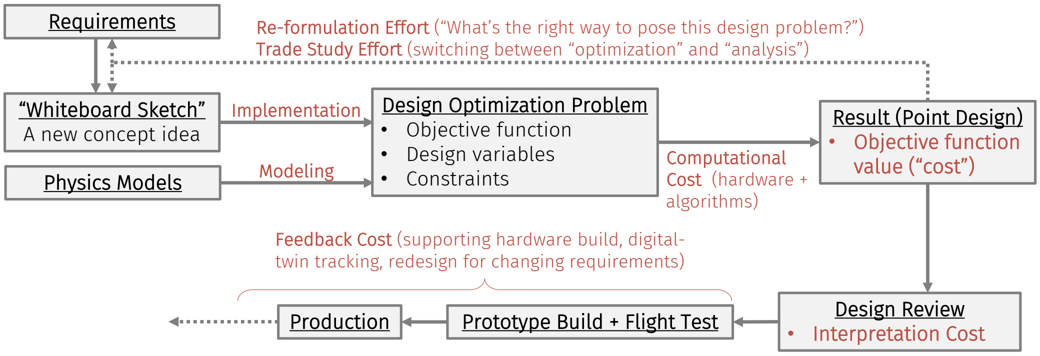
Requirements
Re-formulation Effort ("What's the right way to pose this design problem?")
Trade Study Effort (switching between "optimization" and "analysis")
"Whiteboard Sketch"
A new concept idea
Implementation
Design Optimization Problem
• Objective function
• Design variables
• Constraints
Result (Point Design)
• Objective function value ("cost")
Physics Models
Modeling
Computational Cost (hardware + algorithms)
Feedback Cost (supporting hardware build, digital-twin tracking, redesign for changing requirements)
Production
Prototype Build + Flight Test
Design Review
• Interpretation Cost

Figure 2.2: A high-level overview of the optimization-driven design process, as applied to aircraft design. "Costs" (user frictions to be minimized) associated with each step are shown in red.

In the aircraft design process model of Figure 2.2, the initial point of departure is a set of high-level requirements [29, 36]. The first design step is to develop a set of candidate concepts ("sketches"), a qualitative process that largely leverages designer creativity and experience [35, 37]. Concepts and the associated requirements are then translated into a design optimization problem consisting of an objective function, design variables, and constraints.

The remainder of this section will discuss some of the most important advances in design optimization research that have helped to reduce the costs incurred at various steps of this holistic process.

2.3.1 Two Directions: “Wide” vs. “Deep” MDO

2.3.2 Advances in Optimization Algorithms

Gradient-Based Methods with Analytical Gradients

Disciplined Optimization Methods

2.3.3 Advances in Design Optimization Practice

A notable area where computational methods for aircraft design have long made inroads to industry is in inverse design tools. (As an aerodynamic example, the inverse approach used by the XFoil airfoil design code [39] and others [26] to recover a shape from a specified pressure distribution.) These tools can be seen as an alternative workflow to optimization approaches[6], in that they aim to create a mapping from “performance specification space” back to the “design space”. Drela [7] and others

[6] Interestingly, inverse design tools can be viewed mathematically as a subset of nonlinear optimization, where the feasible space is a single point and there is no objective function.

Chapter 3

Thesis Approach, Methodology, and Contributions

3.1 The Technical Gap: Practical Limitations of MDO Frameworks in Industry

The stark dichotomy of MDO – its immense potential to improve the aircraft design process contrasted with its barriers to practical adoption – is the fundamental motivation for this thesis. The goal of this work is to identify and address the root causes of this gap between academic research and practical industry use. In particular, we will focus on the challenges of *interpretability* and *reviewability* of large-scale engineering design optimization problems. These challenges are not unique to MDO, but rather are a fundamental challenge of large-scale computational tools in general. The field of aircraft design optimization is no exception to this, and it is one that has been recognized for decades. However, the problem remains unsolved.

The primary remaining explanation, which has become increasingly accepted in the past decade [2, 13, 29, 33], is that MDO faces a set of related organizational, culture, and practical challenges that must be addressed before widespread industry adoption can occur.

Summarizing this body of work yields a list of MDO's current major challenges:

1. MDO can be time-consuming to set up. As stated by Torenbeek in 2013 [29] on contemporaneous MDO tools, "the MDO methodology can

be used at any design stage although its complexity does not justify application in the early conceptual phase."

2. MDO can be time-consuming to run [33].

3. MDO faces interpretability challenges, particularly as problem size grows. If MDO is not interpretable and

4. If practitioners are not experienced, it is easy to inadvertently produce designs with poor off-design performance, overly aggressive margins, and violated assumptions [7, 29, 40]. This often leads to non-credible designs or program failures.

5. Most MDO frameworks are syntactically complex, requiring users to be joint experts not only in aircraft design, but also in applied math and computer science [2, 32, 33]. This "expertise barrier", as termed by Grant [41], is a severe impediment to widespread adoption among "users whose focus is the application".

TODO

Interestingly, many of the challenges identified by early MDO works [1, 3] are the same ones that the field is grappling with today.

[32]

TODO

Contemporaneous publications from leading voices in the field of aircraft design around the time of Kroo's work paint a similar story - expressing optimism about the field's growing potential while highlighting (or in some cases, urging) hesitation by industry practitioners.

3.2 Proposed Contributions

The Ph.D. thesis will make the following novel conceptual contributions to address the technical gap described in Section 3.1:

1. **Code Transformation Paradigm:** Introduce a new computational paradigm for MDO frameworks that offers most of the benefits of state-of-the-art paradigms with significantly fewer user frictions.

2. **Differentiable Physics Models:** Provide the first implementations of several key aerospace physics models that are amenable to code transformations.

3. **Sparsity Tracing via NaN-Propagation:** Introduce the new idea of "NaN-propagation" as a method to trace sparsity through black-box numerical analyses, accelerating gradient-based optimization.

4. **Strategies for Black-Box Functions:** Demonstrate that physics-informed machine learning surrogates provide a viable method of incorporating aerospace-related black box functions into a code-transformation-based design framework.

5. **Framework Requirements for Interpretability:** Identify the set of features that a design optimization framework must provide to enable the *interpretability* and *reviewability* of large-scale engineering design optimization problems. Implement these features in a *code transformation* design optimization framework.

6. **System Identification**

7. Build a computational framework for engineering design optimization based on the paradigm of *code transformations*, which enables the use of modern techniques in computer science and applied math (detailed more in Section TODO). This framework will act as a "proving ground" on top of which subsequent research objectives will be implemented and evaluated.

8. Demonstrate that this *code transformation* paradigm enables the formulation of large-scale engineering design optimization problems at the conceptual stage, to an extent that is not otherwise possible with

existing public aircraft design frameworks. To satisfy this objective, the framework will demonstrate (on at least one applied aircraft design case study) performance equaling-or-exceeding the state-of-the-art across the following metrics:

- Runtime speed (i.e., scalability of computational resources, and runtimes compatible with human-in-the-loop interactive design).
- Problem implementation and re-implementation speed (i.e., scalability of engineering time resources).
- Mathematical flexibility: the ability to achieve the above metrics without imposing restrictions on user models that can force undue deviations from physical reality (e.g., log-convexity).

9. Demonstrate the double-edged sword of the large-scale conceptual design optimization capabilities enabled by this framework:

    (a) **Benefit:** Demonstrate that enabling large-scale engineering design at the conceptual stage offers measurable performance gains in the resulting aircraft designs compared to those produced with existing methods. Specifically, the thesis will present at least one applied aircraft design case study, which is then solved with two separate approaches:

    i. A first-order sizing study where an aircraft is designed by coupling only "core" aircraft design disciplines (aerodynamics, structures, and propulsion) and traditional sizing relations (e.g., W/S and T/W diagrams)

    ii. A higher-order sizing study where more "non-core" disciplines (e.g., stability and control, trajectory and mission optimization, cost analysis, field lengths, noise, manufacturability) are added to the aircraft design problem. Geometric flexibility, as measured by the number of design variables describing the aircraft geometry, will also be increased.

    The resulting designs from these two studies will then be compared (post-optimality) to assess their performance on both stated objectives (to assess how much large-scale modeling expands the feasible design space) and secondary metrics (to assess how well core disciplines act as a surrogate for non-core disciplines).

   (b) **Risk:** Demonstrate the major pitfall of the new large-scale conceptual design optimization: the lack of result *interpretability*. Show that the system complexity enabled by large-scale optimization leads to results are more difficult to communicate, interpret, review, and trust.

10.

11. Identify framework features and characteristics that aid in the *interpretability* and *reviewability* of large-scale engineering design optimization problems. This should be based on a combination of literature review, aircraft design case studies, and (as resources allow) discussions with framework users and aircraft designers. Fundamentally, this is more a communication question than a technical one: setting up a large-scale optimization problem is doable with state-of-the-art aircraft design optimization tools, but TODO ("how do we communicate whether we can trust the results of a design optimization study?").

12. Implement some of these practical This should include, at minimum:

    - A constraint activity log, which

3.2.1 Code Transformation Paradigm

The thesis will introduce the idea of *code transformations* as a new computational paradigm on top of which an MDO framework can be built. Here, code transformations are defined as a generalized set of computational techniques that intercept the original optimization problem posed by the user (at runtime), apply some improvement based on analysis of the code itself, and then solve a modified optimization problem instead. This encompasses a variety of recent advanced techniques in scientific computing, such as:

- Automatic differentiation [42]
- Automatic sparsity detection [43]
- Problem transformations:
    - Auto-scaling [44]
    - Log-transformation of variables, constraints, and objectives (similar to geometric programming) [45, 46]
    - Redundant constraint elimination

- Backend-agnostic programming (CPU/GPU, different math library backends, just-in-time compilation, automatic vectorization and parallelization) [47]

The benefit of introducing this abstraction is to recognize that all of these advanced techniques essentially share one major requirement to use: the optimization framework must be able to inspect the actual code driving the design problem. This code inspection is done either via direct source code analysis or by creating a computational graph of the optimization model at runtime. The latter approach, which is called *tracing* in machine learning literature [47–49], is widely preferred to the former in modern, syntactically-rich languages[1]. Therefore, for the purposes of this work, *traceability* is effectively synonymous with code transformability.

If code transformation techniques can be applied to engineering design optimization, they offer order-of-magnitude speedups over the black-box optimization methods that form the vast majority of industry use today [34, 51]. These achievable speeds are comparable to those of state-of-the-art optimization methods in academia, such as disciplined optimization methods[2] [33, 41, 46, 52] and gradient-based methods using user-provided analytic gradients[3] [53–55].

However, code transformations offer a key advantage over these state-of-the-art methods in that they can be applied *automatically* – most of the benefits of these advanced techniques can be gained without additional effort (or mathematical expertise) from the user. This ease-of-use is a critical requirement to overcome the "expertise barrier" described by Grant [41]. Table 3.1 compares code transformations to existing MDO paradigms across three key practical metrics. In short, code transformations offer the best of both worlds: the computational speed of latest academic methods with the ease-of-use of methods already accepted by industry today.

[1] Maclaurin provides a comprehensive discussion of the tradeoffs between these techniques, and why tracing is often preferable in practice [50]

[2] such as geometric programming and disciplined convex programming

[3] sometimes referred to as "adjoint methods" in reference to a common method for manually deriving these gradients for more-complex analyses

Table 3.1: A subjective comparison of tradeoffs between existing MDO framework paradigms and the proposed *code transformation* paradigm. The industrial state-of-the-art is largely *black-box optimization*. The academic state-of-the-art has two major branches: *gradient-based methods with analytical gradients*, and *disciplined optimization methods*. More detailed discussion of these assessments, including definitions and reasoning, is given in Appendix A.

MDO Framework Paradigm	**Example Frameworks and Tools**	**Ease of Implementation**	**Computational Speed and Scalability**	**Mathematical Flexibility**
Black-box Optimization	SUAVE [56], OpenMDAO* [53], TASOPT [9], Most industry codes today	**Best**	**Poor**	**Best**
Gradient-based with Analytic Gradients	MACH-Aero [57], OpenMDAO* [53],	**Poor**	**Best**	**Great**
Disciplined Optimization	GPKit [33], Most algebraic modeling languages	**Good**	**Good**	**Poor**
Code Transformations	AeroSandbox† [58], JAX‡ [47], ModelingToolkit.jl‡ [59]	**Great**	**Great**	**Good**

* Can use either paradigm, depending on user implementation
† Part of the present work, as detailed in Section 4.1
‡ These are computational tools to facilitate code transformations, rather than frameworks themselves

This leads to a compelling value proposition: if we can develop new methods for industry engineers to easily write traceable design code, then we gain access to an host of advanced scientific computing techniques that can significantly shrink the academia-industry MDO gap described in Section 3.1.

Thus, this contribution is twofold. First, the thesis will conceptually introduce code transformations as a new paradigm for engineering MDO frameworks. Secondly, the thesis will demonstrate strategies that allow traceability of engineering design code with minimal user effort, enabling the use of code transformations in practice.

3.2.2 Differentiable Physics Models

The thesis will also contribute the first computational implementations of several key physics models for aircraft design that are compatible with a code transformations paradigm (i.e., traceable).

The broad motivation stems from the observation that "ease-of-implementation" has historically proven to be one of the most important factors for determining whether an MDO paradigm can achieve use in industry. More specifically, the goals of this contribution are to:

1. Stress-test the feasibility of code transformations in practice – How much added user effort and expertise is required to bring typical engineering analyses into a code-transformations-based MDO tool? To what extent can existing code be used as-is? Finally, we aim to identify any specific computational elements that cause "pain points" when attempting to make an analysis traceable.

2. Jump-start future applied research by providing a set of modular, plug-and-play analyses that can be used to quickly build a variety of aircraft design optimization problems. Since the long-term goal of this research direction is to establish whether the proposed MDO paradigm improves practicality, it follows that many practical aircraft design problems must be posed. By creating the building blocks for such problems, we aim to accelerate future research.

To achieve these goals, the thesis will contribute, at minimum, first implementations of the following common aircraft design analyses within a traceable framework. This minimal set of analyses was deliberately chosen to be diverse in terms of numerical considerations, aiming to stress-test a wide variety of engineering code patterns.

Table 3.2

Model	Non-Traceable Analogue	Tests tracing through...
Vortex-Lattice Method	AVL [60]	A geometry engine and discretization; large, vectorized matrix methods, like linear solves.
Nonlinear and Quasi-Linear Lifting Line	Phillips & Snyder [61], Reid [62]	Nonlinear systems of equations (i.e., implicit), which cannot directly be reduced to a static computational graph.
Workbook-style Aerodynamics Buildup	USAF Digital DATCOM [63]	Tests tracing table lookups, performance of tracing on scalar-heavy code
6-DoF Euler-Bernoulli Beam Model	ASWING (structures) [64]	Better
Rigid-Body Equations of Motion		Tests traceability through differential equations, which have unique challenges in stiff ODEs.
Linearized Stability Mode Decomposition	AVL [60]	Tests traceability through more advanced matrix methods, such as an eigenvalue decomposition

3.2.3 Sparsity Tracing via NaN-Propagation

3.2.4 Strategies for Black-Box Functions

3.2.5 Identifying Framework Requirements for Interpretability

An important distinction is that the primary goal of this thesis is not to create the single “end-all” framework for aircraft conceptual MDO - these frameworks naturally come and go over the years as programming languages and optimization algorithms evolve. Rather, the overarching goal is about guiding framework architectures: “What features and paradigms should framework designers consider to mitigate the interpretability challenges of large-scale engineering design optimization?”.

3.2.6 Aircraft System Identification from Minimal Sensor Data

Chapter 4

Status and Proposed Schedule

4.1 Results to Date

4.1.1 Code Transformation Paradigm

In the adjacent field of machine learning, application of these techniques[1] have ignited a revolution of progress [49]. Rackauckas in 2021 [65] states: "[Automatic differentiation] has become the pervasive backbone behind all of the machine learning libraries."

In machine learning, traceability has largely been achieved through the rise of domain-specific languages (DSLs): a sub-language implemented within a programming language. These DSLs, such as Theano, TensorFlow, PyTorch, JAX, and others, have quickly become ubiquitous, empirically demonstrating that DSLs are a viable path to industry adoption. However, there are several downsides if machine-learning-oriented DSLs are applied to engineering design as-is. DSLs tend to differ substantially in syntax from their underlying language, which adds a barrier for new users, and more importantly, requires existing engineering analysis code to be rewritten. DSLs for machine learning also make architectural choices that tend to be ill-suited for typical engineering problems. For example, most deep learning applications consist predominantly of highly-vectorized mathematical operations on large tensors, with a high amount of compute per operator. Engineering analysis, by contrast, tends to be much more scalar-heavy, with many more intermediate steps and a lower

[1] in particular, automatic differentiation and backend-agnostic computing enabling GPU acceleration

amount of compute per operator.

These DSLs restrict users to a narrow set of mathematical operators

Before that, algebraic modeling languages (AMLs) in the field of mathematical optimization

Which is typically achieved through type-flexible programming (through either dynamic typing or multiple dispatch)

4.2 Fields of Study and Coursework

At the first meeting with the thesis committee (Apr. 5, 2023), the following major and minor programs of study were proposed:

- Major: **Computation for Design and Optimization**
  - 16.920: Numerical Methods for Partial Differential Equations
  - 6.255: Optimization Methods
  - 16.888: Multidisciplinary Design Optimization
  - 18.337: Parallel Computing and Scientific Machine Learning
  - 16.842: Fundamentals of Systems Engineering
- Minor: **Flight Physics**
  - 16.110: Flight Vehicle Aerodynamics
  - 16.13: Aerodynamics of Viscous Fluids
  - 16.885: Aircraft Systems Engineering
- Doctoral math requirement:
  - 16.920: Numerical Methods for Partial Differential Equations
  - 6.255: Optimization Methods

The program of study was approved by the committee at the Apr. 2023 meeting, with no further coursework recommendations made. All listed classes have been completed for credit with A/A+ grades, satisfying requirements.

4.3 Degree Milestones

Major degree milestones, both past and future, are listed in Table 4.1.

Table 4.1: Tentative planned timeline for PhD degree milestones.

Complete?	Date	Milestone
✓	Fall 2019	Began studies at MIT
✓	Summer 2021	S.M. thesis submitted
✓	Summer 2021	S.M. degree awarded
✓	Fall 2022	Formation of doctoral committee
✓	Spring 2023	Committee Meeting #1
	Fall 2023	Ph.D. Thesis Proposal Defense
	Spring 2024	Committee Meeting #2
	Fall 2024	Committee Meeting #3
	Winter 2024/Spring 2025	Ph.D. Thesis Defense

4.4 Research Schedule

Spring 2024

1. Motivate the optimization framework developed in the thesis.

   (a) Identify or develop an aircraft design benchmark problem suitable for comparing existing frameworks and their associated paradigms. The design problem should be realistic (incorporating all core aircraft disciplines), but it should not be overly complex. Potential suitable problems could include:

   - SimPleAC, a small aircraft design problem initially proposed by Warren Hoburg [12] and further refined by Berk Ozturk [66].
   - The solar seaplane design problem developed and implemented as part of TA work for MIT 16.821 in Spring 2023 [67].
   - An aircraft design problem adapted from the AIAA Graduate Aircraft Design Competition, where the 2023-24 problem is a self-launching electric sailplane.
   - Long-haul liquid-hydrogen-powered transport aircraft design, which was studied and developed for MIT 16.886 in Fall 2022 and MIT 16.885 in Fall 2023 [68, 69].

   (b) Implement the benchmark problem in several frameworks that are representative of state-of-the-art and/or industry-standard paradigms: AeroSandbox, OpenMDAO, GPKit, and a "optimizer wrapped around black-box analysis" method[2].

   (c) Compare the implementations on the basis of the three framework-level metrics identified in Section 3.2: runtime speed, problem implementation speed, and mathematical flexibility. Document this thoroughly. Use this exercise as a jumping-off point to write a thesis chapter on how specific framework design choices affect each of these three metrics.

   (d) Show that the thesis framework makes large-scale optimization practical to implement at the conceptual design stage.

[2] Typical of industry methods, such as those seen in the development of the Facebook HALE aircraft development effort[23]

2. Quantitatively illustrate the performance benefit associated with large-scale optimization in conceptual-level aircraft design:

   (a) Identify or develop an aircraft design benchmark problem suitable for comparing various levels of model fidelity at the conceptual design level. The design problem should be sufficiently complex such that it can be solved with or without the inclusion of secondary disciplines. Potential suitable problems could include:

      - MIT Firefly, a rocket-propelled micro-UAV design problem coupling packaging constraints, transonic aerodynamics, trajectory design, and propulsion design.
      - Dawn One, a high-altitude-long-endurance solar-powered aircraft.

Fall 2024

1. Present the idea that implicit model assumptions and constraint-satisfaction information as “metadata” that should be

4.5 Facilities Required

No special facilities are required for this work beyond those already available at MIT AeroAstro.

Bibliography

[1] I. Kroo, "Multidisciplinary Optimization Applications in Preliminary Design - Status and Directions," in *38th Structures, Structural Dynamics, and Materials Conference*, (Kissimmee, Florida), American Institute of Aeronautics and Astronautics, Apr. 1997. `https://arc.aiaa.org/doi/10.2514/6.1997-1408`.

[2] J. Agte, O. De Weck, J. Sobieszczanski-Sobieski, P. Arendsen, A. Morris, and M. Spieck, "MDO: Assessment and direction for advancement—an opinion of one international group," *Structural and Multidisciplinary Optimization*, vol. 40, pp. 17–33, Jan. 2010. `http://link.springer.com/10.1007/s00158-009-0381-5`.

[3] H. Ashley, "On Making Things the Best-Aeronautical Uses of Optimization," *Journal of Aircraft*, vol. 19, pp. 5–28, Jan. 1982. `https://arc.aiaa.org/doi/10.2514/3.57350`.

[4] N. Radovcich and D. Layton, "The F-22 structural/aeroelastic design process with MDO examples," in *7th AIAA/USAF/NASA/ISSMO Symposium on Multidisciplinary Analysis and Optimization*, (St. Louis,MO,U.S.A.), American Institute of Aeronautics and Astronautics, Sept. 1998. `https://arc.aiaa.org/doi/10.2514/6.1998-4732`.

[5] J. H. McMasters and R. M. Cummings, "Airplane Design - Past, Present, and Future," *Journal of Aircraft*, vol. 39, pp. 10–17, Jan. 2002. `https://arc.aiaa.org/doi/10.2514/2.2919`.

[6] T. Haftka, "Multidisciplinary Aerospace Design Optimization: Survey of Recent Developments," *Springer, Structural Optimization*, Aug. 1997.

[7] M. Drela, *Pros & Cons of Airfoil Optimization*, pp. 363–381. World

Scientific, Nov. 1998. http://www.worldscientific.com/doi/abs/10.1142/9789812815774_0019.

[8] M. Drela, "Development of the D8 Transport Configuration," in *29th AIAA Applied Aerodynamics Conference*, (Honolulu, Hawaii), American Institute of Aeronautics and Astronautics, June 2011. https://arc.aiaa.org/doi/10.2514/6.2011-3970.

[9] M. Drela, "TASOPT: Transport Aircraft System Optimization. Technical Description." http://web.mit.edu/drela/Public/web/tasopt/TASOPT_doc.pdf, Mar. 2010.

[10] A. Yildirim, J. S. Gray, C. A. Mader, and J. R. Martins, "Performance Analysis of Optimized STARC-ABL Designs Across the Entire Mission Profile," in *AIAA Scitech 2021 Forum*, (VIRTUAL EVENT), American Institute of Aeronautics and Astronautics, Jan. 2021. https://arc.aiaa.org/doi/10.2514/6.2021-0891.

[11] N. Bons, Joaquim R. R. A. Martins, Charles A. Mader, M. McMullen, and M. Suen, "High-fidelity Aerostructural Optimization Studies of the Aerion AS2 Supersonic Business Jet," in *AIAA Aviation 2020 Forum*, 2020.

[12] W. Hoburg and P. Abbeel, "Geometric Programming for Aircraft Design Optimization," *AIAA Journal*, vol. 52, pp. 2414–2426, Nov. 2014. https://arc.aiaa.org/doi/10.2514/1.J052732.

[13] A. Gazaix, F. Gallard, V. Gachelin, T. Druot, S. Grihon, V. Ambert, D. Guénot, R. Lafage, C. Vanaret, B. Pauwels, N. Bartoli, T. Lefebvre, P. Sarouille, N. Desfachelles, J. Brézillon, M. Hamadi, and S. Gurol, "Towards the Industrialization of New MDO Methodologies and Tools for Aircraft Design," in *18th AIAA/ISSMO Multidisciplinary Analysis and Optimization Conference*, (Denver, Colorado), American Institute of Aeronautics and Astronautics, June 2017. https://arc.aiaa.org/doi/10.2514/6.2017-3149.

[14] A. M. Stoll, E. V. Stilson, J. Bevirt, and P. P. Pei, "Conceptual Design of the Joby S2 Electric VTOL PAV," in *14th AIAA Aviation Technology, Integration, and Operations Conference*, (Atlanta,

GA), American Institute of Aeronautics and Astronautics, June 2014. https://arc.aiaa.org/doi/10.2514/6.2014-2407.

[15] T. S. Tao, *Design and Development of a High-Altitude, In-Flight-Deployable Micro-UAV*. S.M., Massachusetts Institute of Technology, 2012. https://dspace.mit.edu/handle/1721.1/76168.

[16] K. Pitta, "Using NAS-developed tools to quiet the boom of supersonic flight." https://www.nas.nasa.gov/pubs/stories/2021/feature_X-59.html, March 2021. Accessed 10-25-2023.

[17] M. Nemec and M. Aftosmis, "Minimizing sonic boom through simulation-based design: The X-59 airplane." https://www.nas.nasa.gov/SC19/demos/demo20.html, November 2019. Accessed 10-25-2023.

[18] H. Mason, "Digital design, multi-material structures enable a quieter supersonic NASA X-plane." https://www.compositesworld.com/articles/digital-design-multi-material-structures-enable-a-quieter-supersonic-nasa-x-plane, May 2022. Accessed 10-25-2023.

[19] S. Choi, J. J. Alonso, I. M. Kroo, and M. Wintzer, "Multifidelity Design Optimization of Low-Boom Supersonic Jets," *Journal of Aircraft*, vol. 45, pp. 106–118, Jan. 2008. https://arc.aiaa.org/doi/10.2514/1.28948.

[20] Z. Lovering, "Vahana configuration trade study – part I." https://acubed.airbus.com/blog/vahana/vahana-configuration-trade-study-part-i/, 12 2016. Accessed 11-03-2023.

[21] G. Bower, "Vahana configuration trade study – part II." https://acubed.airbus.com/blog/vahana/vahana-configuration-trade-study-part-ii/, 2 2017. Accessed 11-03-2023.

[22] "Vahana trade study." https://github.com/VahanaOpenSource/vahanaTradeStudy, 2017. GitHub repository. Accessed 11-03-2023.

[23] N. Roberts, V. Samuel, and D. Colas, "FBHALE." https://github.com/facebookarchive/FBHALE/wiki, 2018. Github repository.

[24] B. Ozturk, W. Hoburg, N. Burnell, and C. Karcher, "Jungle Hawk Owl." https://github.com/convexengineering/jho, 2016. GitHub repository.

[25] S. Wakayama, “Blended-wing-body optimization setup,” in *8th AIAA/USAF/NASA/ISSMO Symposium on Multidisciplinary Analysis and Optimization*, (Long Beach, CA), 2000.

[26] R. Liebeck, M. Page, and B. Rawdon, “Blended-Wing-Body Subsonic Commercial Transport,” in *36th AIAA Aerospace Sciences Meeting and Exhibit*, (Reno,NV,U.S.A.), American Institute of Aeronautics and Astronautics, Jan. 1998. `https://arc.aiaa.org/doi/10.2514/6.1998-438`.

[27] R. Liebeck, “Design of blended wing body subsonic transport,” *Journal of Aircraft*, pp. 10–25, January-February 2004.

[28] G. Vanderplaats, “Automated optimization techniques for aircraft synthesis,” in *Aircraft Systems and Technology Meeting*, (Dallas,TX,U.S.A.), American Institute of Aeronautics and Astronautics, Sept. 1976. `https://arc.aiaa.org/doi/10.2514/6.1976-909`.

[29] E. Torenbeek, *Advanced Aircraft Design: Conceptual Design, Analysis, and Optimization of Subsonic Civil Airplanes.* Aerospace Series, Chichester: Wiley, 2013.

[30] R. Heldenfels, “Automating the design process - Progress, problems, prospects, potential,” in *14th Structures, Structural Dynamics, and Materials Conference*, (Williamsburg,VA,U.S.A.), American Institute of Aeronautics and Astronautics, Mar. 1973. `https://arc.aiaa.org/doi/10.2514/6.1973-410`.

[31] R. R. Heldenfels, “Automation of the Aircraft Design Process,” in *International Council of the Aeronautical Sciences Congress*, (Haifa, Israel), NASA, Aug. 1974. `https://www.icas.org/ICAS_ARCHIVE/ICAS1974/Page%20617%20Heldenfels.pdf`.

[32] A. Salas and J. Townsend, “Framework Requirements for MDO Application Development,” in *7th AIAA/USAF/NASA/ISSMO Symposium on Multidisciplinary Analysis and Optimization*, (St. Louis,MO,U.S.A.), American Institute of Aeronautics and Astronautics, Sept. 1998. `https://arc.aiaa.org/doi/10.2514/6.1998-4740`.

[33] E. Burnell, N. B. Damen, and W. Hoburg, “GPkit: A human-centered approach to convex optimization in engineering design,” in *Proceedings*

of the 2020 CHI Conference on Human Factors in Computing Systems, 2020.

[34] J. R. R. A. Martins and A. Ning, *Engineering Design Optimization*. Cambridge University Press, 1 ed., Nov. 2021. `https://www.cambridge.org/core/product/identifier/9781108980647/type/book`.

[35] M. C. Yang, “Observations on concept generation and sketching in engineering design,” *Research in Engineering Design*, vol. 20, pp. 1–11, Mar. 2009. `http://link.springer.com/10.1007/s00163-008-0055-0`.

[36] E. Torenbeek, *Synthesis of Subsonic Aircraft Design*. Delft: Delft University Press, 1976.

[37] J. Roskam, *Airplane Design*. Ottawa, Kan.: Roskam Aviation and Engineering Corp., 1989.

[38] L. M. Nicolai and G. Carichner, *Fundamentals of Aircraft and Airship Design. Volume 1: Aircraft Design*, vol. 1 of *AIAA Educational Series*. Reston, VA: American Institute of Aeronautics and Astronautics, 2010.

[39] M. Drela, “XFOIL: An Analysis and Design System for Low Reynolds Number Airfoils,” in *Low Reynolds Number Aerodynamics* (C. A. Brebbia, S. A. Orszag, and T. J. Mueller, eds.), vol. 54, (Indiana, USA), pp. 1–12, Springer, 1989. `http://link.springer.com/10.1007/978-3-642-84010-4_1`.

[40] B. Öztürk and A. Saab, “Optimal Aircraft Design Decisions Under Uncertainty Using Robust Signomial Programming,” *AIAA Journal*, vol. 59, pp. 1773–1785, May 2021. `https://arc.aiaa.org/doi/10.2514/1.J058724`.

[41] M. Grant, S. Boyd, and Y. Ye, “Disciplined Convex Programming,” in *Global Optimization* (L. Liberti and N. Maculan, eds.), vol. 84, pp. 155–210, Boston: Kluwer Academic Publishers, 2006. `http://link.springer.com/10.1007/0-387-30528-9_7`.

[42] A. Griewank, “On Automatic Differentiation,” p. 28, 1988.

[43] A. H. Gebremedhin, A. Tarafdar, A. Pothen, and A. Walther, "Efficient Computation of Sparse Hessians Using Coloring and Automatic Differentiation," *INFORMS Journal on Computing*, vol. 21, pp. 209–223, May 2009. https://pubsonline.informs.org/doi/10.1287/ijoc.1080.0286.

[44] J. Nocedal and S. J. Wright, *Numerical Optimization.* Springer Series in Operations Research, New York: Springer, 2nd ed ed., 2006.

[45] P. Kirschen and W. Hoburg, "The power of log transformation: A comparison of geometric and signomial programming with general nonlinear programming techniques for aircraft design optimization," 2018.

[46] A. Agrawal, S. Diamond, and S. Boyd, "Disciplined geometric programming," *Optimization Letters*, vol. 13, pp. 961–976, July 2019. http://link.springer.com/10.1007/s11590-019-01422-z.

[47] J. Bradbury, R. Frostig, P. Hawkins, M. J. Johnson, C. Leary, D. Maclaurin, G. Necula, A. Paszke, J. VanderPlas, S. Wanderman-Milne, and Q. Zhang, "JAX: composable transformations of Python+NumPy programs," 2018.

[48] R. Frostig, M. J. Johnson, and C. Leary, "Compiling machine learning programs via high-level tracing," in *SYSML '18*, (Stanford, CA, USA), Feb. 2018. https://mlsys.org/Conferences/2019/doc/2018/146.pdf.

[49] A. G. Baydin, B. A. Pearlmutter, A. A. Radul, and J. M. Siskind, "Automatic differentiation in machine learning: A survey." http://arxiv.org/abs/1502.05767, Feb. 2018.

[50] D. Maclaurin, *Modeling, Inference, and Optimization With Composable Differentiable Procedures.* PhD thesis, Harvard University, Cambridge, Massachusetts, 2016. https://dash.harvard.edu/bitstream/handle/1/33493599/MACLAURIN-DISSERTATION-2016.pdf.

[51] A. Lavin, D. Krakauer, H. Zenil, J. Gottschlich, T. Mattson, J. Brehmer, A. Anandkumar, S. Choudry, K. Rocki, A. G. Baydin, C. Prunkl, B. Paige, O. Isayev, E. Peterson, P. L. McMahon, J. Macke, K. Cranmer, J. Zhang, H. Wainwright, A. Hanuka, M. Veloso, S. Assefa, S. Zheng, and A. Pfeffer, "Simulation Intelligence: Towards a New Generation of Scientific Methods." http://arxiv.org/abs/2112.03235, Nov. 2022.

[52] S. P. Boyd and L. Vandenberghe, *Convex Optimization.* Cambridge, UK ; New York: Cambridge University Press, 2004.

[53] J. S. Gray, J. T. Hwang, J. R. R. A. Martins, K. T. Moore, and B. A. Naylor, "OpenMDAO: An open-source framework for multidisciplinary design, analysis, and optimization," *Structural and Multidisciplinary Optimization*, vol. 59, pp. 1075–1104, Apr. 2019. `http://link.springer.com/10.1007/s00158-019-02211-z`.

[54] G. K. Kenway, C. A. Mader, P. He, and J. R. Martins, "Effective adjoint approaches for computational fluid dynamics," *Progress in Aerospace Sciences*, vol. 110, p. 100542, Oct. 2019. `https://linkinghub.elsevier.com/retrieve/pii/S0376042119300120`.

[55] M. Innes, "Don't Unroll Adjoint: Differentiating SSA-Form Programs," *arXiv:1810.07951 [cs]*, Mar. 2019. `http://arxiv.org/abs/1810.07951`.

[56] T. MacDonald, M. Clarke, E. M. Botero, J. M. Vegh, and J. J. Alonso, *SUAVE: An Open-Source Environment Enabling Multi-Fidelity Vehicle Optimization.*

[57] P. He, C. A. Mader, J. R. Martins, and K. J. Maki, "An aerodynamic design optimization framework using a discrete adjoint approach with OpenFOAM," *Computers & Fluids*, vol. 168, pp. 285–303, May 2018. `https://linkinghub.elsevier.com/retrieve/pii/S0045793018302020`.

[58] P. D. Sharpe, "AeroSandbox: A Differentiable Framework for Aircraft Design Optimization," Master's thesis, Massachusetts Institute of Technology, 2021.

[59] Y. Ma, S. Gowda, R. Anantharaman, C. Laughman, V. Shah, and C. Rackauckas, "ModelingToolkit: A Composable Graph Transformation System for Equation-Based Modeling," *arXiv:2103.05244 (cs)*, 2021.

[60] M. Drela and H. Youngren, "AVL: Athena vortex lattice." `https://web.mit.edu/drela/Public/web/avl/`, 1989–2023. Accessed 11-09-2023.

[61] W. F. Phillips and D. O. Snyder, "Modern Adaptation of Prandtl's Classic Lifting-Line Theory," *Journal of Aircraft*, vol. 37, pp. 662–670, July 2000. `https://arc.aiaa.org/doi/10.2514/2.2649`.

[62] J. T. Reid, *A General Approach to Lifting-Line Theory, Applied to Wings with Sweep.* PhD thesis, Utah State University, Aug. 2020. https://digitalcommons.usu.edu/cgi/viewcontent.cgi?article=8982&context=etd.

[63] R. Fink, "USAF Stability and Control DATCOM," Tech. Rep. AFWAL-TR-83-3048, McDonnell Douglas Corporation, Douglas Aircraft Division, 1960. for the Flight Controls Division, Air Force Flight Dynamics Laboratory, Wright-Patterson AFB, Ohio. October 1960, revised November 1965, revised April 1978.

[64] M. Drela, "ASWING user guide." https://web.mit.edu/drela/Public/web/aswing/, 2012. Accessed: 11-09-2023.

[65] C. Rackauckas, "Generalizing Automatic Differentiation to Automatic Sparsity, Uncertainty, Stability, and Parallelism." https://www.stochasticlifestyle.com/generalizing-automatic-differentiation-to-automatic-sparsity-uncertainty-stability-and-parallelism/, Mar. 2021.

[66] B. Ozturk, *Conceptual Engineering Design and Optimization Methodologies Using Geometric Programming.* PhD thesis, Massachusetts Institute of Technology, 2018.

[67] P. Sharpe, "Solar seaplane preliminary sizing." https://github.com/peterdsharpe/solar-seaplane-preliminary-sizing, 2023. GitHub repository.

[68] J. Gaubatz, E. Martin, A. Miyamoto, B. Murga, P. Sharpe, M. Travnik, A. Tsay, Z. J. Wang, and R. J. Hansman, "Estimating the Energy Demand of a Hydrogen-Based Long-Haul Air Transportation Network," in *2023 International Conference on Future Energy Solutions (FES)*, (Vaasa, Finland), pp. 1–6, IEEE, June 2023. https://ieeexplore.ieee.org/document/10182543/.

[69] P. Sharpe, "Transport aircraft sizing." https://github.com/peterdsharpe/transport-aircraft, 2023. Github repository.

Appendix A

Comparison of MDO Framework Paradigms