

Geometric Programming for Aircraft Design Optimization

Warren Hoburg* and Pieter Abbeel†
 University of California, Berkeley, Berkeley, California, 94720

We propose formulating conceptual-stage aircraft design problems as geometric programs (GPs), which are a specific type of convex optimization problem. Recent advances in convex optimization offer significant advantages over the general nonlinear optimization methods typically used in aircraft design. Modern GP solvers are extremely fast, even on large problems, require no initial guesses or tuning of solver parameters, and guarantee globally optimal solutions. These benefits come at a price: all objective and constraint functions – the mathematical models that describe aircraft design relations – must be expressed within the restricted functional forms of GP. Perhaps surprisingly, this restricted set of functional forms appears again and again in prevailing physics-based models for aircraft systems. Moreover, we show that for various models that cannot be manipulated algebraically into the forms required by GP, we can often fit compact GP models which accurately approximate the original models. The speed and reliability of GP solution methods makes them a promising approach for conceptual-stage aircraft design problems.

Nomenclature

A	=	aspect ratio	S	=	wing area [m ²]
b	=	wing span [m]	S_r	=	root section modulus [m ³]
C_D	=	total drag coefficient	T	=	thrust force [N]
C_f	=	skin friction coefficient	\bar{t}_{cap}	=	spar cap thickness per unit chord
C_L	=	lift coefficient	\bar{t}_{web}	=	shear web thickness per unit chord
C_{Dp}	=	profile drag coefficient	\mathbf{u}	=	vector of all decision variables
CDA_0	=	non-wing drag area [m ²]	V	=	flight speed [m/s]
D	=	drag force [N]	V_{stall}	=	stall speed, flaps down [m/s]
e	=	Oswald efficiency factor	W	=	operating weight [N]
g	=	gravitational constant, 9.8m/s ²	W_{cap}	=	spar cap weight [N]
h_{fuel}	=	fuel heating value [J/kg]	W_{eng}	=	engine weight [N]
\bar{h}_{rms}	=	root mean square spar box height	$W_{\text{fuel,out}}$	=	fuel burned, outbound [N]
I_r	=	root area moment of inertia [m ⁴]	$W_{\text{fuel,ret}}$	=	fuel burned, return [N]
\bar{I}_{cap}	=	area moment of inertia per chord ⁴	W_{MTO}	=	maximum takeoff weight [N]
k	=	pressure drag form factor	W_{pay}	=	payload weight [N]
L	=	lift force [N]	W_{web}	=	shear web weight [N]
L'	=	lift force per unit span [N/m]	W_{wing}	=	wing weight [N]
M_r	=	root moment [N·m]	W_{zfw}	=	zero fuel weight [N]
\dot{m}_{fuel}	=	fuel mass flow rate [kg/s]	\tilde{W}	=	weight excluding wing [N]
\bar{M}_r	=	root moment per chord, M_r/c_r	x_{TO}	=	takeoff distance [m]
N_{lift}	=	ultimate load factor	z	=	helper variable
p	=	$\equiv 1 + 2\lambda$	z_{bre}	=	Breguet parameter
P_{fuel}	=	fuel power, $\dot{m}_{\text{fuel}}h_{\text{fuel}}$ [W]	δ	=	wing tip deflection [m]
P_{max}	=	max engine output power [W]	η	=	non-dimensional spanwise coordinate
q	=	$\equiv 1 + \lambda$	η_0	=	overall efficiency
R	=	range [m]	η_i	=	inviscid propeller efficiency
Re	=	Reynolds number	η_v	=	viscous propeller efficiency
			η_{eng}	=	engine efficiency
			η_{prop}	=	propeller efficiency
			λ	=	wing taper ratio
			ν	=	$\equiv (1 + \lambda + \lambda^2)/(1 + \lambda)^2$
			ρ	=	air density [kg/m ³]
			τ	=	wing thickness ratio
			θ_{fuel}	=	fuel fraction, $W_{\text{fuel}}/W_{\text{zfw}}$
			ξ	=	takeoff drag-to-thrust ratio

I. Introduction

Over the past 50 years, computation and numerical simulation have become standard tools in aircraft design. These advances have opened new doors for system-wide optimization, creating the possibility of quantifying and exploiting tradeoffs to create more finely tuned flying machines.

Efforts to combine analysis models into a coordinated optimization framework show great promise, but also continue to uncover serious computational and organizational challenges [1]. Multidisciplinary design optimization (MDO) has become a research field with significant economic potential and interest from a wide range of stake-holders. Numerous MDO formulations have emerged, with various breakdowns of computation into subproblems, communication schemes for passing

*Ph.D. Candidate, Department of Electrical Engineering and Computer Science, whoburg@alum.mit.edu, AIAA Member.

†Assistant Professor, Department of Electrical Engineering and Computer Science, pabbeel@cs.berkeley.edu.

coupling variables among computational blocks, and varied degrees of feasibility enforced after each iteration [2–8]. There has been significant progress on Kriging and response surface methods that reduce or eliminate calls to expensive analysis codes [9, 10]. Reduced order modeling is another area of interest, where complex models are replaced by cheaper surrogates that still capture relevant input/output relationships [11, 12]. In high-fidelity optimization, *adjoint methods* have enabled efficient computation of gradients in PDE-constrained optimization. This line of research started in the 1970’s with Pironneau’s study of control theory applied to shape optimization [13, 14], and was followed by Jameson’s seminal work on aerodynamic design [15, 16]. These methods have received significant research attention [17–19], and are the state of the art for high-fidelity aero-structural optimization [20, 21]. Finally, there is recent interest in *multi-fidelity* methods [22], where inexpensive models could guide which high-fidelity analyses to conduct.

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. Most design tools in commercial or industrial use today solve specific problems involving a few disciplines (such as an engine design subroutine or a coupled aero-structural wing optimization). Tools for optimizing systems of black-box computer simulations often take a long time (days, weeks, or more) to arrive at a single solution.

When an aircraft configuration is first evaluated, the goal is to understand tradeoffs among various facets of the aircraft and mission. In many cases, the objectives (e.g. design missions) are not even defined, so the goal is to understand the shape of a Pareto frontier and guide early program decision making or identify business opportunities. The range of models involved, along with the need to solve many similar design problems to sweep out trade studies, calls for reliable and efficient optimization methods that provide robust approximations over a wide range of parameter inputs.

Reliable and efficient optimization is difficult for general optimization problems, but may be possible for more specialized problems. Over the past two decades, applied mathematicians have transformed *convex optimization* from a specialized research discipline into the realm of a technology [23]. Like solving least-squares problems or linear programs^a, solving standard classes of convex optimization problems exactly is a straightforward task for modern solvers. Recently, an increasing number of engineering disciplines have begun utilizing and relying upon this new technology [24–26]. That said, convex optimization is notably absent from most MDO approaches, with the exception of sequential quadratic programming (SQP) methods for solving non-convex optimization problems locally.

In this work, we show that a specialized type of convex optimization problem, the geometric program (GP), applies to a surprisingly wide range of physics-based models common in conceptual stage aircraft design. We also show that GP-compatible functions can approximate models that are not analytical, such as data generated by black box simulation.

The GP approach is not universal. By restricting ourselves to special functional forms, we give up the ability to model arbitrary nonlinear relationships. In return, we get something extremely powerful. Unlike solving a general nonlinear optimization problem, which is hard, solving a GP is fast and easy. Modern solvers find *globally* optimal solutions, with fast solution times that scale to large problems.

Compared with MDO, the methods of this article represent a unique approach. In both cases, the ultimate goal is to efficiently arrive at solutions that are supported by accurate modeling. Much of MDO starts with extremely accurate *models*, and makes sacrifices in efficiency or quality of optimization (e.g., accepting a local instead of global optimum). In contrast, the proposed approach starts with extremely reliable and efficient *optimization*, and makes sacrifices in the accuracy or fidelity of the models one can optimize over. The unique efficiency and reliability of GP methods makes them a powerful tool for optimizing large, multidisciplinary systems of low-order models.

^aBoth least-squares problems and linear programs are, in fact, special cases of convex optimization problems.

II. Geometric Programming

First introduced in 1967 by Duffin, Peterson, and Zener [27], a GP is a specific type of constrained optimization problem that becomes convex after a logarithmic change of variables. Despite significant work on early applications in structural design [28], network flow [29], and optimal control [30, 31], reliable and efficient numerical methods for solving GPs were not available until the 1990’s [32]. GP has recently undergone a resurgence as researchers discover promising applications in digital circuit design [24], communication systems [25], antenna optimization [26], and statistics [23].

A. Geometric Program: Definition

This section uses *power law notation*: for two vectors $\mathbf{u}, \mathbf{a} \in \mathbb{R}^n$,

$$\mathbf{u}^{\mathbf{a}} \equiv \prod_{j=1}^n u_j^{a_j}. \quad (1)$$

Geometric programs are constrained optimization problems where the objective and constraints consist only of *monomial* and *posynomial* functions. Let us begin by defining these two special function classes.

1. Monomial Functions

In geometric programming, a *monomial*^b is a function^c $h(\mathbf{u}) : \mathbf{u} \in \mathbb{R}_{++}^n \rightarrow \mathbb{R}_{++}$ of the form

$$h(\mathbf{u}) = c\mathbf{u}^{\mathbf{a}}, \quad (2)$$

where $\mathbf{a} \in \mathbb{R}^n$, and $c \in \mathbb{R}_{++}$. For instance, the familiar expression for lift, $\frac{1}{2}\rho V^2 C_L S$, is a monomial in $\mathbf{u} = (\rho, V, C_L, S)$, with $c = 1/2$ and $\mathbf{a} = (1, 2, 1, 1)$. Since the powers a_i in (2) may be negative and non-integer, expressions like $\frac{u_1 u_2^{0.7} \sqrt{u_3}}{u_4}$ are also monomials. The property $\forall \mathbf{u}, h(\mathbf{u}) > 0$ holds for any monomial function h .

2. Posynomial Functions

Like monomials, *posynomials* are functions $f(\mathbf{u}) : \mathbf{u} \in \mathbb{R}_{++}^n \rightarrow \mathbb{R}_{++}$. A posynomial has the form

$$f(\mathbf{u}) = \sum_{k=1}^K c_k \mathbf{u}^{\mathbf{a}_k}, \quad (3)$$

where (as before) $\mathbf{a}_k \in \mathbb{R}^n$, and $c_k \in \mathbb{R}_{++}$. Thus, a posynomial is simply a sum of monomial terms, and all monomials are also posynomials (with just one term). The expression $0.23 + u_1^2 + 0.1u_1u_2^{0.8}$ is an example of a posynomial in $\mathbf{u} = (u_1, u_2)$, whereas $2u_1 - u_2^{1.5}$ is not a posynomial because negative leading coefficients c_k are not allowed.

3. Geometric Program in Standard Form

A *geometric program in standard form* (also called a GP in posynomial form) is a non-linear, non-convex optimization problem of the form

$$\begin{aligned} &\text{minimize} && f_0(\mathbf{u}) \\ &\text{subject to} && f_i(\mathbf{u}) \leq 1, \quad i = 1, \dots, m, \\ &&& h_i(\mathbf{u}) = 1, \quad i = 1, \dots, m_e, \end{aligned} \quad (4)$$

where the f_i are posynomial (or monomial) functions, the h_i are monomial functions, and $\mathbf{u} \in \mathbb{R}_{++}^n$ are the *decision variables*. In plain English, a GP minimizes a posynomial objective function, subject to monomial equality constraints and posynomial inequality constraints. Monomials and posynomials are both closed under monomial division, so constraints of the form (posynomial \leq monomial) or (monomial = monomial) are easily converted into the form in (4). Also, any monomial equality constraint $h(\mathbf{u}) = 1$ may be expressed equivalently as two

^bAs noted in [33], the term *monomial* carries a special meaning in GP; the term used in algebra is slightly different.

^c \mathbb{R}_{++} represents the strictly positive reals.

monomial inequality constraints: $h(\mathbf{u}) \leq 1$ and $1/h(\mathbf{u}) \leq 1$. Thus without loss of generality, a geometric program in standard form can always be written in the inequality-constrained form

$$\begin{aligned} & \text{minimize} && \sum_{k=1}^{K_0} c_{0k} \mathbf{u}^{a_{0k}} \\ & \text{subject to} && \sum_{k=1}^{K_i} c_{ik} \mathbf{u}^{a_{ik}} \leq 1, \quad i = 1, \dots, m. \end{aligned} \quad (5)$$

This form is expected by some modern commercial solvers. The objective and constraints contain a combined total of $t = \sum_{i=0}^m K_i$ monomial terms. The entire GP is parameterized by a vector of constants $\mathbf{c} \in \mathbb{R}_{++}^t$, an (often sparse) matrix of exponents $\mathbf{A} \in \mathbb{R}^{t \times n}$, and a mapping that encodes which of the $m + 1$ posynomials each of the t monomial terms resides in.

B. Solving Geometric Programs

Over the past two decades, technology for solving GPs has become extremely reliable and efficient. At their core, today's state-of-the-art solvers implement *primal-dual interior point methods* [32, 34]. When applied to GPs, these methods provide remarkable capabilities.

Optimality GP solvers guarantee convergence to a *global* optimum (or a certificate of infeasibility, if it is impossible to simultaneously satisfy all the constraints).

Robustness GP solvers perform off-the-shelf. They do not require 'initial guesses' or tuning of optimizer hyper-parameters.

Speed Current solvers are approaching the efficiency of linear program (LP) solvers: they can find the global optimum of a problem with thousands of decision variables and tens of thousands of constraints on a desktop computer in minutes [23], with additional gains if the problem is sparse. As of 2005, a typical *sparse* GP with tens of thousands of decision variables and one million constraints could be solved on a desktop computer in a few minutes [33].

Strong Duality Primal-dual interior point algorithms simultaneously determine globally-optimal *dual* variables, in addition to the primal variables. These dual variables provide useful sensitivity information [33].

The power of geometric programming derives from a change of variables that converts GPs into *convex optimization problems*. GPs are not convex in the form (4), but a simple logarithmic change of variables transforms monomials into affine functions, and posynomials into log-sum-exp functions, both of which are convex [33]. The resulting solution performance contrasts sharply with methods for general non-linear optimization, which typically require initial guesses, may require the tuning of problem-specific optimizer parameters, and find local, not global, optima. GP solvers, on the other hand, are robust and general enough for users to confidently leave the optimization process to standard software packages.

III. A Simple Example

As an initial warm-up to build familiarity with GP formulations, we present a simple wing design example adapted from the course materials of the class *Introduction to MDO* at Stanford [35]. A more complex design example will appear in Section VI.

Our challenge is to size a wing with total area S , span b , and aspect ratio $A = b^2/S$. These parameters should be chosen to minimize the total drag, $D = \frac{1}{2}\rho V^2 C_D S$. The drag coefficient is modeled as the sum of fuselage parasite drag, wing parasite drag, and induced drag,

$$C_D = \frac{(CDA_0)}{S} + kC_f \frac{S_{\text{wet}}}{S} + \frac{C_L^2}{\pi Ae}, \quad (6)$$

where (CDA_0) is the fuselage drag area, k is a form factor that accounts for pressure drag, S_{wet}/S is the wetted area ratio, and e is the Oswald

efficiency factor. For a fully turbulent boundary layer, the skin friction coefficient C_f can be approximated as

$$C_f = \frac{0.074}{\text{Re}^{0.2}}, \quad (7)$$

where $\text{Re} = \frac{\rho V}{\mu} \sqrt{\frac{S}{A}}$ is the Reynolds number at the mean chord $c = \sqrt{S/A}$. The total aircraft weight W is modeled as the sum of a fixed weight W_0 and the wing weight,

$$W = W_0 + W_w. \quad (8)$$

The wing weight is modeled as

$$W_w = 45.42S + 8.71 \times 10^{-5} \frac{N_{\text{lift}} b^3 \sqrt{W_0 W}}{S \tau}, \quad (9)$$

where N_{lift} is the ultimate load factor for structural sizing, and τ is the airfoil thickness to chord ratio. The weight equations are coupled to the drag equations by the constraint that lift equals weight,

$$W = \frac{1}{2} \rho V^2 C_L S. \quad (10)$$

Finally, for safe landing, the aircraft should be capable of flying at a reduced speed V_{min} , subject to a stall constraint,

$$\frac{2W}{\rho V_{\text{min}}^2 S} \leq C_{L,\text{max}}. \quad (11)$$

Table 1. Fixed constants for the simple example problem in Section III.

Quantity	Value	Units	Description
(CDA_0)	0.0306	m ²	fuselage drag area
ρ	1.23	kg/m ³	density of air
μ	1.78×10^{-5}	kg/(ms)	viscosity of air
S_{wet}/S	2.05		wetted area ratio
k	1.2		form factor
e	0.96		Oswald efficiency factor
W_0	4940	N	aircraft weight excluding wing
N_{lift}	2.5		ultimate load factor
τ	0.12		airfoil thickness to chord ratio
V_{min}	22	m/s	desired landing speed
$C_{L,\text{max}}$	2.0		max C_L , flaps down

We must choose values of S , A , and V that minimize drag, subject to all the relations in the preceding text. Constant parameters are given in Table 1.

For this problem, it turns out that the *global* optimum can be found reliably on a laptop computer in a few milliseconds. The key is recognizing that all the models consist of monomial and posynomial expressions. In fact, the entire optimization problem can be expressed exactly

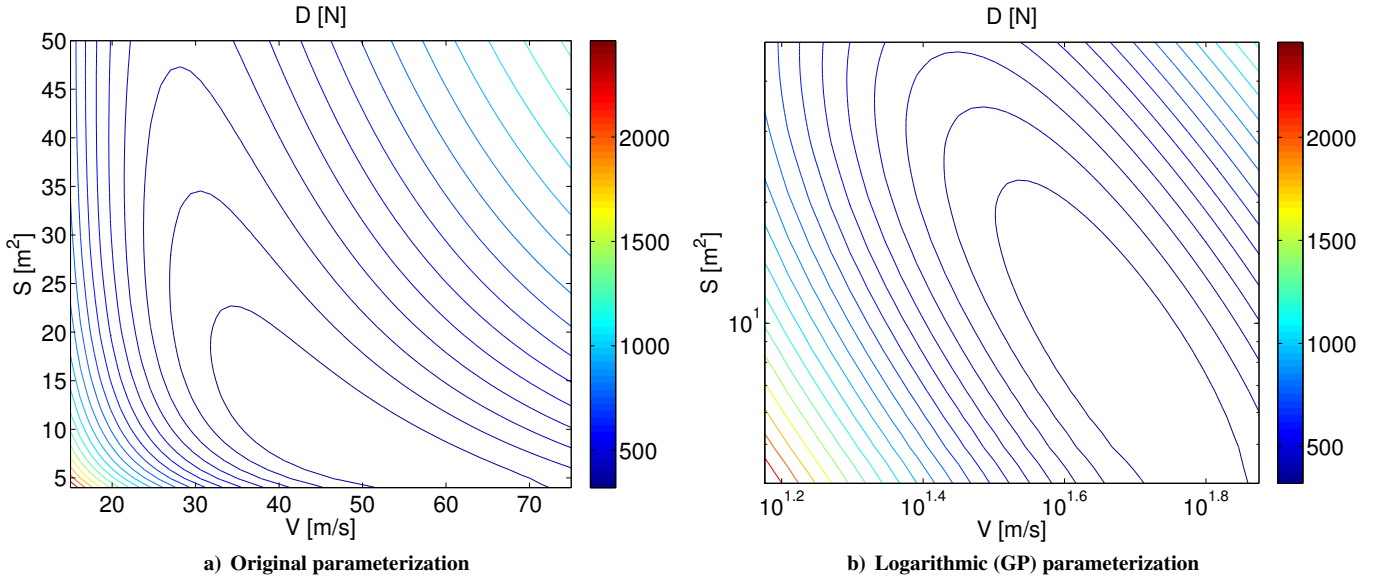


Figure 1. Slice of the design space for the simple example problem. The contours represent the drag objective as a function of two of the design variables: cruise velocity and wing area. In the original parameterization, the contours are not convex, and are not well-approximated by level sets of a quadratic function. This implies that iterative Newton descent algorithms (SQP, for example) could require many iterations to converge (especially if the starting point is not well-chosen). In contrast, after the (GP) log transformation, the level sets are convex and well-approximated by quadratic level sets (ellipses).

as a GP^d:

$$\begin{aligned}
 & \underset{A, S, C_D, C_L, C_f, Re, W, W_w, V}{\text{minimize}} && \frac{1}{2} \rho V^2 C_D S && (12) \\
 & \text{subject to} && 1 \geq \frac{0.074}{C_f Re^{0.2}} \\
 & && 1 \geq \frac{(CDA_0)}{C_D S} + \frac{k C_f S_{wet}}{C_D S} + \frac{C_L^2}{C_D \pi A e} \\
 & && 1 \geq \frac{2W}{\rho V^2 C_L S} \\
 & && 1 \geq \frac{W_0}{W} + \frac{W_w}{W} \\
 & && 1 \geq 45.42 \frac{S}{W_w} + 8.71 \times 10^{-5} \frac{N_{lift} A^{3/2} \sqrt{W_0 W S}}{W_w \tau} \\
 & && 1 \geq \frac{2W}{\rho V_{min}^2 S C_{L,max}} \\
 & && 1 = \frac{\rho V}{\mu Re} \sqrt{\frac{S}{A}}
 \end{aligned}$$

For a design engineer wishing to solve this problem, no further algorithmic work is required. A GP solver [36] finds the solution reliably and quickly:

Variable	Optimal Value	Units
A	12.7	
S	12.08	m
C _D	0.0231	
C _L	0.6512	
C _f	0.003857	
Re	2.598e+06	
W	7189	N
W _w	2249	N
V	38.55	m/s

This aircraft design is *feasible*, meaning that it satisfies all the previously defined design constraints. It is also *globally optimal*, meaning

^dTo construct this GP, we used a *posynomial equality relaxation*, described in Section IV.

that no other set of feasible decision variables could possibly achieve a lower value of the objective, $1/2\rho V^2 C_D S$.

Of course, in a real design setting, a single point solution is utterly inadequate. A wise designer or manager would want to understand a range of possible tradeoffs. How would modifying the desired landing speed $V_{min} = 22\text{m/s}$ affect the drag objective? How expensive would it be to fly at a slightly higher cruise speed V than that which minimizes drag? The answer to these questions lies in a *Pareto frontier*, which quantifies the tradeoffs among the relevant variables.

In Fig. 2, we show how GP-based design can be used as a powerful inner loop for quickly exploring Pareto frontiers. For this design example, we re-solved the GP (12) across a range of different landing speeds V_{min} . Then, for each landing speed of interest, we re-solved across a range of different cruise speeds, starting with the drag-optimal V . The resulting tradeoff surface, shown in Fig. 2a, represents the design space of aircraft that are Pareto-optimal with respect to drag, cruise speed, and landing speed.

IV. The GP Design Paradigm

Now that we have seen a specific example, let us understand, more generally, how GP can be applied to solve real design problems.

A. Elements of a GP

1. Decision Variables

Referring to the definition of a GP (4), the *decision variables* are a vector of unknowns $\mathbf{u} \in \mathbb{R}_{++}^n$, implicitly constrained to be positive^e. In the previous example (12), the decision variables were $\mathbf{u} = (A, S, C_D, C_L, C_f, Re, W, W_w, V)$. More generally, the decision variables consist of every variable whose value is to be determined by the optimizer.

Clearly, the decision variables cannot take on arbitrary values; they are constrained by physics. These physical relationships are quantified by *constraints* on the feasible set of the GP.

^eThe restriction $\mathbf{u} > 0$ is not as limiting as one might initially assume. Many quantities (e.g. weight, drag, efficiency, thicknesses, climb or descent angle) are strictly positive (or strictly negative) physical quantities whose value can be captured by a strictly positive decision variable. Variables whose signs are unknown prior to optimization can be handled in a mixed-integer extension to GP.

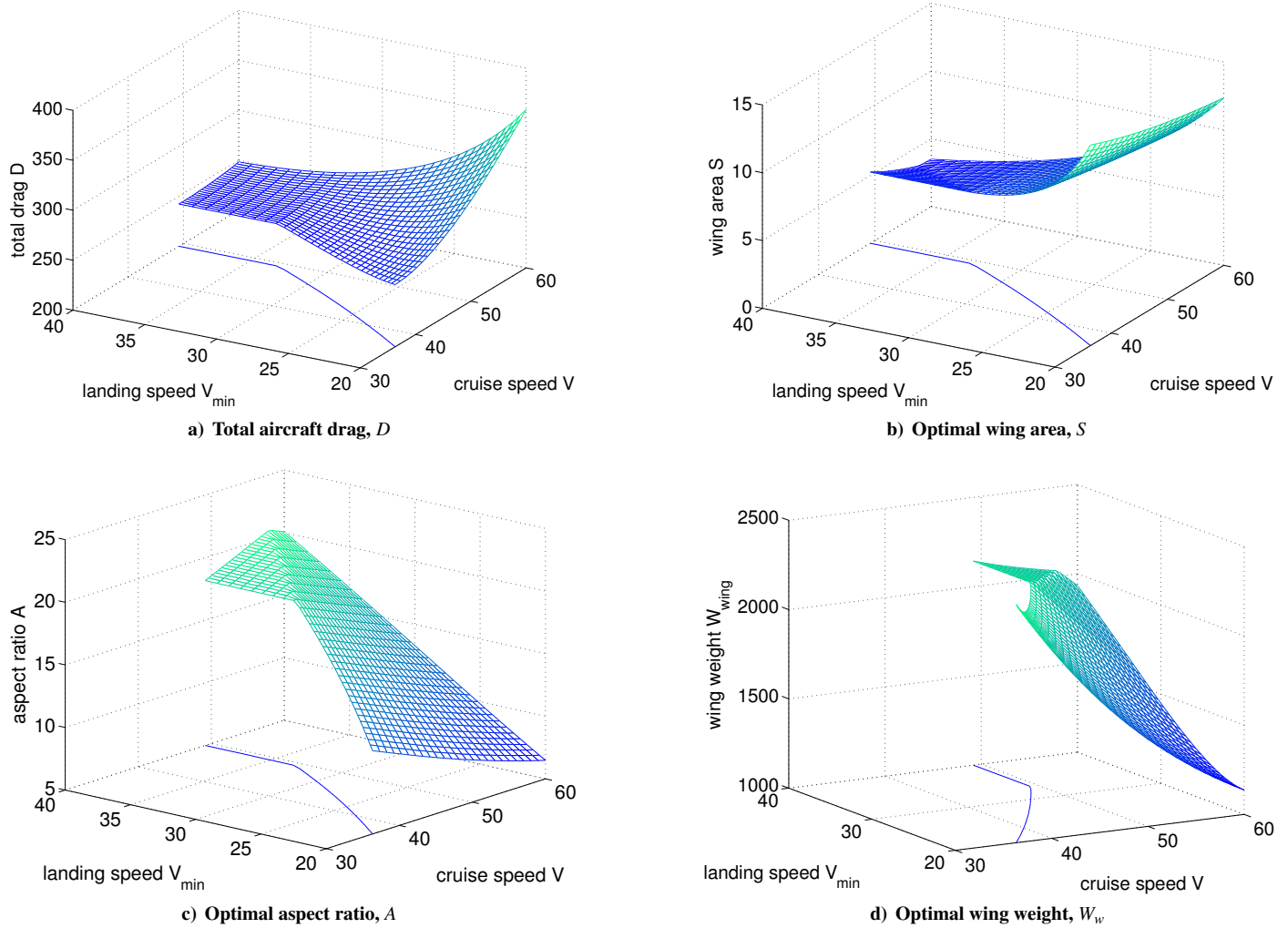


Figure 2. Tradeoff surfaces for the wing design problem in Section III. Here the GP (12) was solved 775 times, across a grid of unique cruise speeds, V , and landing speeds, V_{\min} . This resulted in the Pareto frontier (a), which trades off low cruise drag D , high cruise speed V , and low landing speed V_{\min} . The corresponding optimal design parameters appear in the other figures, where each point on the meshes corresponds to a unique aircraft design. The thin line plotted below each mesh represents the drag-optimal cruise speed as a function of landing speed. On a standard laptop, sweeping out the full Pareto frontier (i.e. solving the GP 775 times) took 3.28 seconds total, or 4.2 milliseconds per solution on average.

2. Constraints

In the context of GP-based design, constraints serve several purposes:

GP Modeling is the process of posing a practical design problem as a GP, by encoding the underlying physics in terms of monomial and posynomial constraints. Examples of GP-compatible aircraft design models appear in Section V.

Engineering limits are constraints that impose practical or manufacturing limits on decision variables such as material gauges, structural stresses, deflections, part sizes, margins of safety, etc.

Requirements are system-level performance bounds imposed by the design team or customer. These constraints are often simple expressions involving a single decision variable, e.g. Range > 5000 km, or $W_{\text{payload}} > 49000$ N.

3. Objective

The objective in a GP may be any monomial or posynomial function of the decision variables. One might choose to globally minimize a single variable, such as drag or fuel weight. To trade off a number of performance criteria, one can form an aggregate objective function – a weighted combination of several quantities of interest. In order to *maximize* quantities such as velocity or efficiency, we can minimize the monomial corresponding to their inverse.

B. Posynomial Equality Relaxation

Posynomial equality relaxation is a GP modeling technique that is central to the GP design paradigm. The basic idea is to relax a posynomial equality constraint into an inequality constraint, thereby making it GP-compatible. Under certain conditions, this relaxation is tight, meaning that equality will hold at the optimum [33].

Consider, for example, a simple model for drag that breaks down C_D into a constant profile drag component, and an induced drag component:

$$C_D = C_{d_0} + \frac{C_L^2}{\pi e A} \quad (13)$$

While we recognize the posynomial structure in this model, it is not GP-compatible, because posynomial *equality* constraints are not allowed in GP. Indeed, enforcing a posynomial equality constraint is in general a very difficult problem. But thanks to our knowledge of the variables involved, we can relax this constraint to an inequality:

$$C_D \geq C_{d_0} + \frac{C_L^2}{\pi e A} \quad (14)$$

Even though we have relaxed the problem, we can show that the original equality relationship (13) will be globally optimal under certain assumptions about the functional behavior of the objective and constraints with respect to C_D . In particular, we assume that C_D does not appear in any monomial equality constraints, and that the objective and inequality constraints are all monotone increasing (or constant) in C_D . Under these conditions, if the equality relation (13) did not hold at the optimum, we could clearly decrease C_D until achieving equality, without increasing the objective or moving the solution outside the feasible set.

This type of relaxation is widely applicable, and can also be applied when the direction of the assumed monotonicities are reversed [31, 33]. We will use it extensively in Section V without further comment.

V. Selected GP-compatible Models

In this section, we give further examples of GP-compatible models for the aircraft design domain. Here *GP-compatible* refers to models of two possible forms:

1. $1 \geq f(\mathbf{u})$, where f is a posynomial (or monomial)
2. $1 = h(\mathbf{u})$, where h is a monomial

Note that monomials and posynomials are closed under monomial division. This implies that models of the following forms are also GP-compatible:

3. $h(\mathbf{u}) \geq f(\mathbf{u})$, where f is a posynomial (or monomial), and h is a monomial
4. $h_1(\mathbf{u}) = h_2(\mathbf{u})$, where h_1 and h_2 are both monomials

Finally, observe throughout this section that each model involves only a subset of the full decision variable vector \mathbf{u} . This means that the resulting GP is *sparse*.

A. Steady Level Flight Relations

The steady level flight relations are perhaps the most basic relationships in aircraft design. These constrain lift to equal the aircraft's total weight, and thrust to equal the force of drag.

$$L = W \quad (15)$$

$$T = D \quad (16)$$

$$L = \frac{1}{2} \rho V^2 C_L S \quad (17)$$

$$D \geq \frac{1}{2} \rho V^2 C_D S \quad (18)$$

Each of these equations are monomial constraints, and therefore GP-compatible.

The total aircraft weight W breaks down into a sum of component weights, for instance

$$W_{zfw} \geq W_{\text{fixed}} + W_{\text{payload}} + W_{\text{wing}} + W_{\text{engine}} + W_{\text{tail}} + \dots, \quad (19)$$

$$W \geq W_{zfw}(1 + \theta_{\text{fuel}}), \quad (20)$$

where $\theta_{\text{fuel}} \equiv \frac{W_{\text{fuel}}}{W_{zfw}}$ is the fuel mass fraction. Each of these is a posynomial inequality constraint, and therefore GP-compatible. Using similar models, further breakdown and modeling of component weights is straightforward.

Another important steady flight relation is the chain of efficiencies η that relate cruise thrust power to fuel power. A simple version is

$$TV \leq P_{\text{fuel}} \eta_{\text{eng}} \eta_{\text{prop}}, \quad (21)$$

where TV is thrust times velocity; $P_{\text{fuel}} = \dot{m}_{\text{fuel}} h_{\text{fuel}}$ is the fuel power, equal to fuel mass flow rate times heating value; η_{eng} is the engine's fuel power to shaft power conversion efficiency, and η_{prop} is the propulsive shaft power to thrust power conversion efficiency.

These constraints capture high-level relationships among lift, drag, weight, and efficiency. Additional constraints will capture their dependency on more detailed design parameters.

B. Performance Metrics

The GP framework provides a straightforward interface for trading off competing goals: we optimize or constrain multiple *performance metrics* of interest.

When a performance metric is also a decision variable, (cruise speed, payload capacity, or fuel burn rate, for example), it can be inserted directly into the objective function or constraints. Other metrics are more complicated summary statistics, whose relationships to other variables must themselves be modeled. In this section, we give two examples: range and takeoff distance.

Breguet Range Equation

One common model for the range of a fuel-burning aircraft is the Breguet range equation [37], which assumes a constant lift to drag ratio L/D and overall efficiency η_0 , resulting in the expression

$$R = \frac{h_{\text{fuel}}}{g} \eta_0 \frac{L}{D} \log(1 + \theta_{\text{fuel}}). \quad (22)$$

This expression is not allowed in GP due to the log term, but we can proceed by rewriting it as

$$1 + \theta_{\text{fuel}} = \exp\left(\frac{gRD}{h_{\text{fuel}}\eta_0 L}\right), \quad (23)$$

and noting that the Taylor expansion of the exp function has a posynomial structure^f. This results in a GP-compatible model:

$$z = \frac{gRD}{h_{\text{fuel}}\eta_0 L} \quad (24)$$

$$\theta_{\text{fuel}} \geq z + \frac{z^2}{2!} + \frac{z^3}{3!} + \dots \quad (25)$$

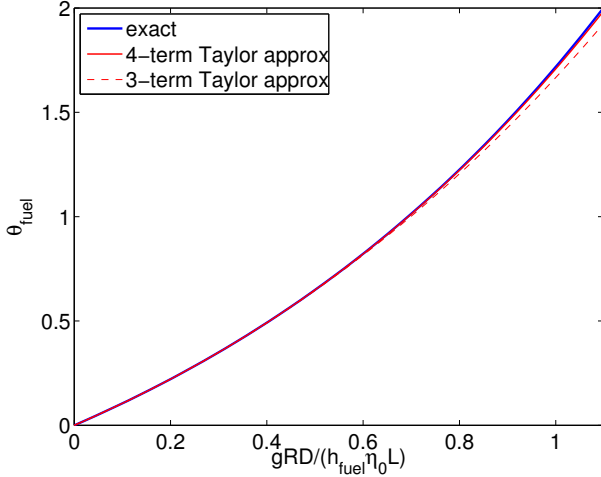


Figure 3. GP-compatible approximation of the Breguet range equation, via posynomial structure in the Taylor expansion of exp. A 3-term expansion is more than 99% accurate for fuel fractions less than 0.95, and a 4-term expansion is more than 99% accurate for fuel fractions less than 2.0.

Recall that the Breguet model assumed that the lift to drag ratio L/D and overall efficiency η_0 remained constant over the entire mission. For real missions, these quantities vary slightly with changes in wing loading, speed, and density altitude. To model these effects more accurately, one can break down long or complex missions into shorter segments of length R_j . One would then constrain each segment according to the Breguet range equation, but allow each segment to take on a different lift to drag ratio L_i/D_i , overall efficiency $\eta_{0,i}$, and fuel fraction $\theta_{\text{fuel},i}$. This approach enables accurate modeling of very long or complex missions, and has the added benefit of reducing the fuel fraction of each segment, thereby improving the accuracy of the Taylor approximation shown in Fig. 3.

Takeoff Distance

To model takeoff distance x_{TO} , we define a wheels-up flight condition immediately after rotation, where the aircraft first achieves lift $L_{\text{TO}} \geq W$, and is still accelerating under thrust $T_{\text{TO}} > D_{\text{TO}}$. Prior to this instant, a net force $T - D$ accelerated the aircraft from speed 0 to speed V_{TO} . Technically both T and D are functions of V , but let us assume that the thrust variation is small, taking the conservative approximation $T(V) = T(V_{\text{TO}}) = T_{\text{TO}}$. Under this assumption, we have the differential relation

$$gdx = \frac{WVdV}{T_{\text{TO}} - \frac{1}{2}\rho V^2 C_D S} \quad (26)$$

^fWe could alternatively treat the exp function directly instead of Taylor expanding it, but at the expense of requiring more specialized convex programming software instead of a standard GP solver.

from basic mechanics. If we additionally assume that C_D stays constant ($C_D(V) = C_D^{\text{TO}}$), then we can analytically integrate (26) along the takeoff run, which results in the expression

$$x_{\text{TO}} = \frac{WV_{\text{TO}}^2}{2gD_{\text{TO}}} \log\left(\frac{T_{\text{TO}}}{T_{\text{TO}} - D_{\text{TO}}}\right). \quad (27)$$

To clarify the limiting behavior as $D_{\text{TO}}/T_{\text{TO}} \rightarrow 0$, we rewrite (27) as

$$\frac{2gx_{\text{TO}}T_{\text{TO}}}{WV_{\text{TO}}^2} = \frac{-\log(1-\xi)}{\xi}, \quad (28)$$

where $\xi \equiv D_{\text{TO}}/T_{\text{TO}}$. This expression is not compatible with GP, but we can proceed by modeling the $-\log(1-\xi)/\xi$ term with a posynomial, as shown in Fig. 4. This results in a set of GP-compatible constraints for takeoff distance:

$$\xi \geq \frac{D_{\text{TO}}}{T_{\text{TO}}} \quad (29)$$

$$\frac{2gx_{\text{TO}}T_{\text{TO}}}{WV_{\text{TO}}^2} \geq 1 + y \quad (30)$$

$$1 \geq \frac{0.0464\xi^{2.73}}{y^{2.88}} + \frac{1.044\xi^{0.296}}{y^{0.049}} \quad (31)$$

Equation (31) is an example of an *implicit posynomial*, so called because it expresses the relationship between y and ξ as a convex set whose boundary is an implicit function $f(y, \xi)$. Implicit posynomials are known to be more expressive than their explicit counterparts [38], and their compatibility with GP makes them a powerful general modeling tool.

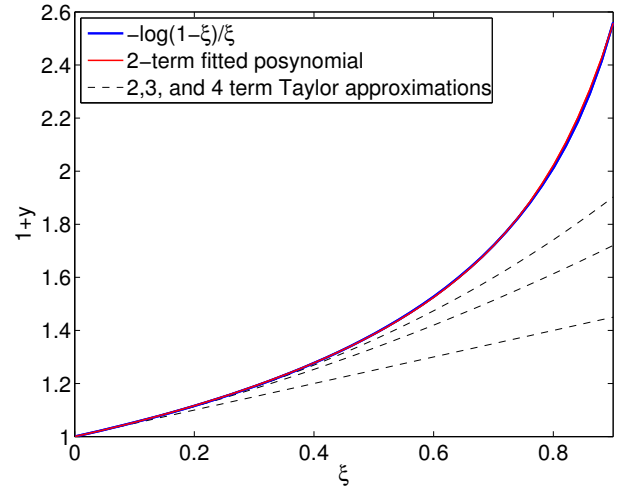


Figure 4. GP-compatible posynomial approximation of the function $-\log(1-\xi)/\xi$, for use in modeling takeoff distance. Taylor expansions (dashed lines) are one option, but would require many terms to achieve a reasonable fit. In contrast, a fitted implicit posynomial (solid curve) achieves a near-perfect fit with only two terms.

As suggested for the Breguet range model, one can optionally refine the accuracy of the takeoff distance model by dividing the takeoff run into multiple individually-modeled segments. We can also include 50' obstacle clearance distance using GP-compatible models for climbing flight.

C. Propulsive Efficiency

A propeller converting mechanical shaft power P_{shaft} into propulsive power TV experiences losses that vary significantly with both thrust and velocity. Following Drela [39], we model propulsive efficiency as the product of a viscous loss term η_v , and an inviscid term η_i that accounts for kinetic energy lost in the high-velocity prop-wash:

$$\eta_{\text{prop}} = \eta_i \eta_v. \quad (32)$$

Actuator disk theory gives us the following approximations for the inviscid efficiency:

$$\dot{m} = \frac{1}{2}\rho A_{\text{prop}}(V_e + V) \quad (33)$$

$$T = \dot{m}(V_e - V) = \frac{1}{2}\rho A_{\text{prop}}(V_e^2 - V^2) \quad (34)$$

$$\eta_i = \frac{TV}{\frac{1}{2}\dot{m}(V_e^2 - V^2)} = \frac{2}{1 + \frac{V_e}{V}} \quad (35)$$

Combining the above, we obtain

$$\eta_i \leq \frac{2}{1 + \sqrt{1 + \frac{T}{\frac{1}{2}\rho V^2 A_{\text{prop}}}}}. \quad (36)$$

The quantity $T/(\frac{1}{2}\rho A_{\text{prop}}V^2)$ is recognized as C_T/λ_a^2 , where C_T is the propeller thrust coefficient, and λ_a is the advance ratio. The constraint (36) is not allowed by the GP framework, but we can algebraically manipulate it into an equivalent posynomial constraint,

$$4\eta_i + \frac{T\eta_i^2}{\frac{1}{2}\rho V^2 A_{\text{prop}}} \leq 4 \quad (37)$$

This GP-compatible model captures the strong variation of propulsive efficiency with thrust and velocity.

D. Drag Breakdown

The total aircraft drag, $D = 1/2\rho V^2 C_D S$, clearly depends on the drag coefficient, C_D , which can in turn be modeled as a sum of contributions from different sources. For example, in subsonic flight regimes, we might break down C_D into three components:

$$C_D \geq \underbrace{\frac{C_L^2}{\pi e A}}_{\text{induced drag}} + \underbrace{C_{Dp}(C_L, \text{Re}, \tau)}_{\text{profile drag}} + \underbrace{\frac{(CDA)_0}{S}}_{\text{non-wing form drag}}. \quad (38)$$

The induced drag term comes from lifting line theory [40], which predicts a vortex-induced downwash distribution over the wing that effectively reduces the angle of attack.

The function $C_{Dp}(C_L, \text{Re}, \tau)$ is not known analytically, and must be fit from data. We can generate this data offline using the viscous airfoil analysis tool XFOIL [41], as shown in Fig. 5. This particular data set is well approximated by a posynomial model, which can be fit offline using nonlinear least-squares [38, 42].

The final term in the drag breakdown corresponds to form drag on the fuselage and other components. For a detailed treatment, it can be further broken up into a posynomial model for component drag areas:

$$(CDA)_0 \geq (CDA)_{\text{tail}} + (CDA)_{\text{fuse}} + (CDA)_{\text{gear}} + \dots, \quad (39)$$

Each term in (39) may either be assumed constant, or modeled as a posynomial function of relevant sizing parameters.

E. Wing Structural Models

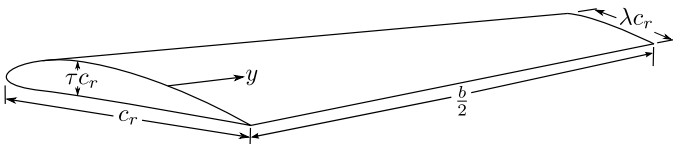


Figure 6. Wing design variables τ (airfoil thickness to chord ratio) and λ (taper ratio), for a single-taper wing or wing section.

In this section we consider the structural design of an un-swept single-taper wing (or tail), as depicted in Fig. 6. The high-level stress constraint we will impose is

$$S_r \sigma_{\text{max}} \geq N_{\text{lift}} M_r \quad (40)$$

where S_r is the root section modulus, σ_{max} is the material-specific allowable stress, N_{lift} is the design G loading or turbulence loading including safety factor, and M_r is the applied moment at the root[§].

We may also wish to impose a deflection limit, e.g.

$$\frac{\delta}{b} \leq 0.05 \quad (41)$$

We must now model the spanwise lift distribution, applied root moment, and bending stiffness.

Coordinate Definitions

The wing sizing variables are related by:

$$b = \sqrt{SA} \quad (42)$$

$$c_r = \frac{2}{1 + \lambda} \sqrt{\frac{S}{A}} \quad (43)$$

We also define a wing spanwise coordinate $2y/b = \eta \in [0, 1]$. The chord as a function of span is then

$$\frac{c(\eta)}{c_r} = 1 + \eta(\lambda - 1). \quad (44)$$

Root Moment

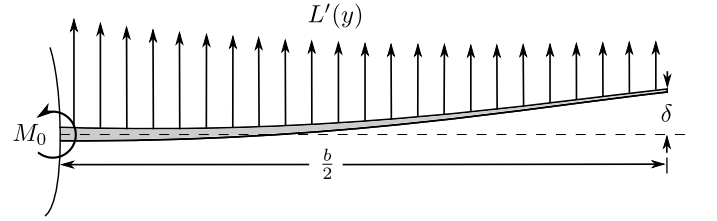


Figure 7. The spanwise lift distribution $L'(y)$ creates a root moment M_0 and tip deflection δ .

The moment applied to the wing root depends on the spanwise lift distribution, as shown in Fig. 7. The wing must support its own weight (as well as any fuel contained in it), in addition to the weight of the fuselage and payload. Let $\tilde{W} = \tilde{L}$ represent the weight of the aircraft excluding the wing. A simple conservative assumption assumes that the local net-upward-force-per-unit-span, \tilde{L}' , is proportional to the local chord [37]. Under this assumption, the differential loading per unit span is

$$d\tilde{L} = \frac{\tilde{L}}{1 + \lambda} [1 + \eta(\lambda - 1)] d\eta \quad (45)$$

To find the root moment, we twice-integrate Equation (45) with appropriate boundary conditions, which results in

$$M_r = \frac{\tilde{L}b}{12} \left[\frac{1 + 2\lambda}{1 + \lambda} \right] = \frac{\tilde{L}Ac_r}{24} (1 + 2\lambda) \quad (46)$$

Root Stiffness

The wing root's ability to resist applied moments is governed by two important quantities: the root area moment of inertia, I_r , and the root section modulus, S_r . S_r relates applied moments to maximum bending stress, whereas I_r relates applied moments to curvature (and therefore, deflection). For a symmetric structural cross section, the two quantities are related by the monomial

[§]We actually impose the constraint (40) for two different values of S_0 : one for tensile (bottom skin) loading, and one for compressive (top skin) loading.

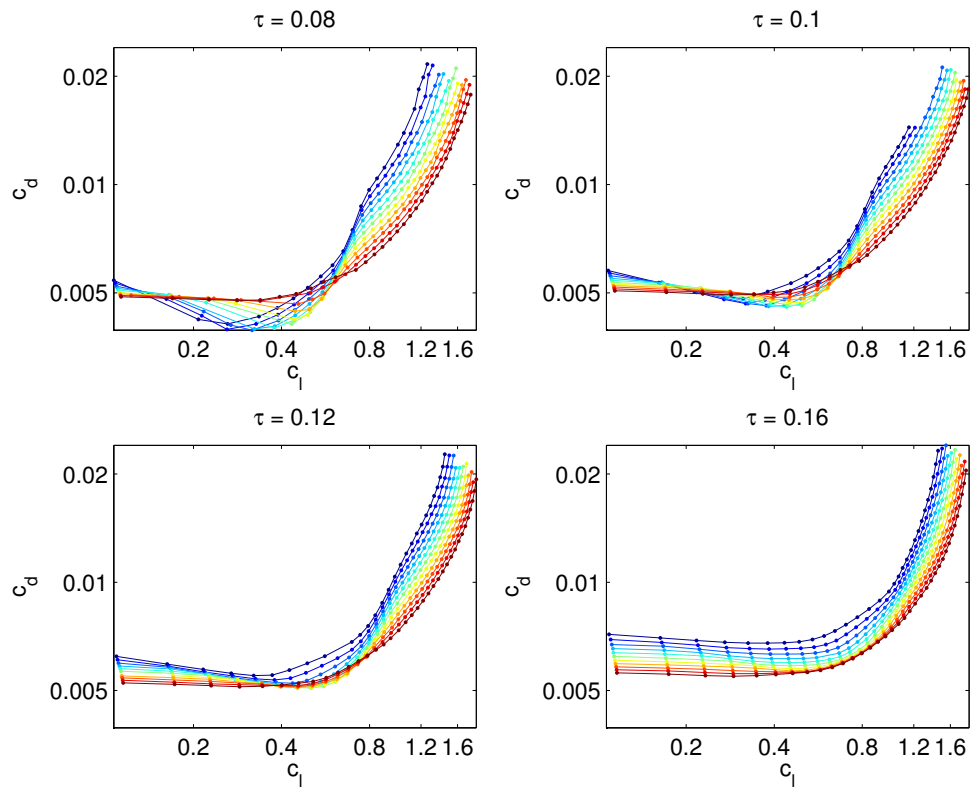


Figure 5. Profile drag data as a function of c_l for NACA-24xx airfoils, plotted here on a log scale for slices at four different values of airfoil thickness τ . Shades represent Reynolds number, which ranged from $Re = 10^5$ (dark blue) to $Re = 10^7$ (dark red). Each line represents a sweep across angle of attack for a particular airfoil at a constant Re (i.e., a type-1 lift-drag polar). Circles indicate the actual data points, which were generated using XFOIL [41]. An implicit posynomial surrogate model with 8 terms approximates this entire data set with an RMS error of approximately 2%.

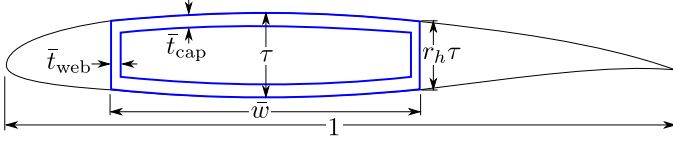


Figure 8. Structural cross section geometry for unit-chord airfoil, from Drela [43]. The structural wing box has the same maximum thickness as the airfoil, τ , and tapers quadratically to a fraction r_h at the webs.

$$S_r = \frac{I_r}{\frac{1}{2}\tau c_r}. \quad (47)$$

It is generally possible to fit a posynomial model for the area moment of inertia per chord to the fourth, $\bar{I} = I/c^4$, for an airfoil family and structural geometry of choice. For example, here we will use a wing box geometry defined by Drela [43] and shown in Fig. 8. The spar cap \bar{I} is related to material thickness by

$$\bar{I}_{\text{cap}} = \frac{\bar{w}}{12} (\bar{h}_{\text{rms}}^3 - (\bar{h}_{\text{rms}} - 2\bar{\tau}_{\text{cap}})^3) \approx \frac{\bar{w}}{2} (\bar{h}_{\text{rms}}^2 \bar{\tau}_{\text{cap}} - 2\bar{h}_{\text{rms}} \bar{\tau}_{\text{cap}}^2), \quad (48)$$

where \bar{h}_{rms} is the root mean square box height. If we assume that $r_h = 0.75$, then $\bar{h}_{\text{rms}} \approx 0.92\tau$, and a posynomial model for \bar{I}_{cap} is

$$0.92\bar{w}\tau\bar{\tau}_{\text{cap}}^2 + \bar{I}_{\text{cap}} \leq \frac{0.92^2}{2} \bar{w}\tau^2 \bar{\tau}_{\text{cap}} \quad (49)$$

Making the conservative assumption that the bending stress is carried by the caps only ($\bar{I}_{\text{cap}} \gg \bar{I}_{\text{web}}$), the stress limit (40) becomes

$$\frac{2\bar{I}_{\text{cap}} c_r^3 \sigma_{\text{max}}}{\tau} \geq N_{\text{lift}} M_r. \quad (50)$$

Shear Web Sizing

Assuming that all shear loads are carried by the web, the root shear stress is

$$\sigma_{\text{shear}} = \frac{\bar{L}}{4c_r^2 \bar{\tau}_{\text{web}} r_h \tau}. \quad (51)$$

Letting $r_h = 3/4$ and substituting (43) for c_r , we obtain the shear web sizing relation:

$$\frac{12\tau S \bar{\tau}_{\text{web}} \sigma_{\text{max, shear}}}{A \bar{L} N_{\text{lift}}} \geq 1 + 2\lambda + \lambda^2 \quad (52)$$

Wing System Component Masses

To determine the weight of the spar caps and shear webs, we must integrate their spanwise area distribution. A wing structural component (\cdot), with area per chord squared $\bar{A}_{\cdot} = A_{\cdot}/c^2$, has a total weight of

$$W_{\cdot} = \rho_c g 2 \int_0^{b/2} \frac{A_{\cdot}(y)}{c(y)^2} \frac{c(y)^2}{c_r^2} c_r^2 dy = \rho_c g \bar{A}_{\cdot} c_r^2 b \int_0^1 \frac{c(\eta)^2}{c_r^2} d\eta \quad (53)$$

The spar cap and shear web areas are

$$\bar{A}_{\text{cap}} = 2\bar{w}\bar{\tau}_{\text{cap}} \quad (54)$$

$$\bar{A}_{\text{web}} = 2r_h \bar{\tau}_{\text{web}} \quad (55)$$

Evaluating the integral (53), we obtain weight equations for the spar caps and shear webs:

$$W_{\text{cap}} = \frac{8\rho_{\text{cap}} g \bar{w} \bar{\tau}_{\text{cap}} S^{3/2}}{3A^{1/2}} \left[\frac{\lambda^2 + \lambda + 1}{(1 + \lambda)^2} \right] \quad (56)$$

$$W_{\text{web}} = \frac{8\rho_{\text{web}} g r_h \bar{\tau}_{\text{web}} S^{3/2}}{3A^{1/2}} \left[\frac{\lambda^2 + \lambda + 1}{(1 + \lambda)^2} \right] \quad (57)$$

Wing Tip Deflection

Under Euler-Bernoulli bending theory, we have the relationship

$$\frac{d^2 \delta}{dy^2} = \frac{M(y)}{EI_{xx}(y)} \quad (58)$$

Because both M and I_{xx} vary with y , integrating (58) can introduce significant complication. One conservative simplifying assumption is that the curvature is constant along the span and equal to the root curvature. This leads to the relationship

$$\delta = \frac{1}{2} \frac{M_r}{EI_r} \left(\frac{b}{2} \right)^2, \quad (59)$$

where E is the Young's modulus of the spar cap.

GP-Compatible Formulation

The key relations for modeling are (46), (49), (50), (52), (56), (57), and (59). At first glance, we note that these equations are not GP-compatible, since they are not all posynomial in λ . To make the wing structural relations GP-compatible, we introduce the change of variables

$$p = 1 + 2\lambda \quad (60)$$

We also define $q = 1 + \lambda$. p and q are related by the posynomial constraint

$$2q \geq 1 + p \quad (61)$$

We now proceed by expressing the governing relations in terms of p and q instead of λ ^b.

We define the root moment per root chord, $\bar{M}_r = M_r/c_r$, and replace (46) with the equivalent monomial constraint

$$\bar{M}_r \geq \frac{\bar{L} A p}{24}. \quad (62)$$

The area moment of inertia model (49) is already a posynomial constraint on \bar{I}_{cap} , and does not require further modification. The stress limit (50) becomes a monomial constraint,

$$8 \geq \frac{N_{\text{lift}} \bar{M}_r A q^2 \tau}{S \bar{I}_{\text{cap}} \sigma_{\text{max}}} \quad (63)$$

The shear web sizing equation (52) also becomes a monomial,

$$12 \geq \frac{A \bar{L} N_{\text{lift}} q^2}{\tau S \bar{\tau}_{\text{web}} \sigma_{\text{max, shear}}} \quad (64)$$

To handle the weight equations (56) and (57), we introduce the function $\nu(\lambda) = (\lambda^2 + \lambda + 1)/(1 + \lambda)^2$, and note that ν is log-convex in p . We can approximate $\nu(p)$ via the posynomial constraint

$$\nu^{3.94} \geq 0.86p^{-2.38} + 0.14p^{0.56}. \quad (65)$$

The approximation error is very close to 0, as shown in Fig. 9. The weight equations then become monomials:

$$W_{\text{cap}} \geq \frac{8\rho_{\text{cap}} g \bar{w} \bar{\tau}_{\text{cap}} S^{3/2} \nu}{3A^{1/2}} \quad (66)$$

$$W_{\text{web}} \geq \frac{8\rho_{\text{web}} g r_h \bar{\tau}_{\text{web}} S^{3/2} \nu}{3A^{1/2}} \quad (67)$$

Finally, under the change of variables, the wing deflection equation (59) is also equivalent to a monomial:

$$\delta \geq \frac{A^{5/2} \bar{M}_r q^3}{64 S^{1/2} E_{\text{cap}} \bar{I}_{\text{cap}}} \quad (68)$$

^b In order for our posynomial equality relaxations to hold with equality, we must ensure that with the exception of (61), all constraints involving q are monotone increasing in q . That is, for all constraints $1 \geq f(q)$ involving q , we need $\partial f/\partial q \geq 0$, which holds for all models presented herein.

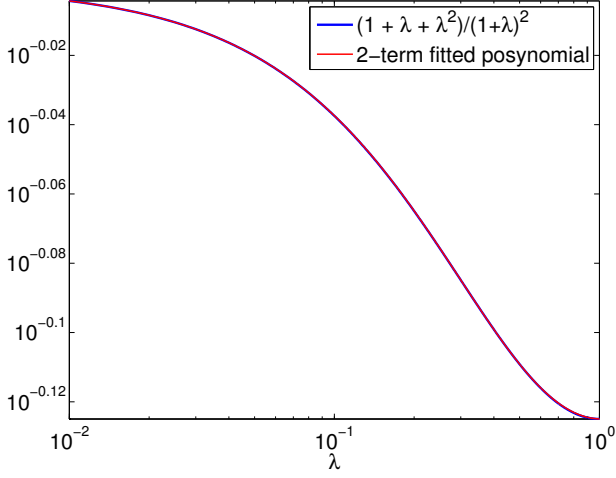


Figure 9. The function $v(\lambda) = (1 + \lambda + \lambda^2)/(1 + \lambda)^2$ is clearly not log-convex, and is therefore not GP-compatible. Nevertheless, the function is *quasi-convex*. This allows us to apply a change of variables (60) that causes v to become log-convex in $p = 1 + 2\lambda$ over the range $1 \leq p \leq 3$, which corresponds to $\lambda \in [0, 1]$. We can then model $v(p)$ with a posynomial (65). The approximation error is less than 0.03%.

The models we have presented so far will favor small values of λ , since tapering the wing has significant structural benefits in the form of weight savings. However, too small a λ can be dangerous, since it can overload the outboard sections of the wing, leading to risk of tip stall. It is therefore prudent to set a lower limit on λ , e.g.

$$p \geq 1.9 \quad (69)$$

VI. Design Example

Here we solve an example design problem via GP-compatible modeling. Our task is the design of a new UAV, which will fly an out-and-back reconnaissance mission.

A. Objective

The objective is to minimize the total out-and-back fuel burn.

B. Requirements

The high-level vehicle requirements are

1. Specified range for out-and-back mission, $R \geq 5000\text{km}$
2. Specified payload for out-and-back mission, $m_{\text{pay}} \geq 500\text{kg}$
3. Sprint speed requirement, separate from design mission, $V_{\text{sprint}} \geq 150\text{m/s}$
4. Stall speed requirement for safe landing after aborted takeoff at MTOW, $V_{\text{stall}} \leq 38\text{m/s}$

C. Propulsion

The vehicle will be powered by a single turboprop engine, with propulsive efficiency governed by (37). We assume a power-law scaling for engine weight as a function of installed power [44].

D. Design Mission

The vehicle must fly out and back some specified distance, at a cruise altitude of 3000m. For GP modeling we split this mission into two legs, outbound and return, with different flight conditions (velocity, lift and drag coefficients, efficiency, etc.) along each leg. The fuel burn along each leg is governed by the Breguet range equation. Climb and

descent are ignored for simplicity, although we note that this framework is entirely capable of far more detailed climb, cruise, and descent analysis.

Table 2. Fixed constant parameters for the design problem in Section VI.

Quantity	Value	Units	Description
N_{lift}	6.0		wing loading multiplier
σ_{max}	250×10^6	Pa	allowable stress, 6061-T6
$\sigma_{\text{max, shear}}$	167×10^6	Pa	allowable shear stress
ρ_{alum}	2700	kg/m^3	aluminum density
g	9.8	m/s^2	gravitational constant
\bar{w}	0.5		wing box width/chord
r_h	0.75		see Fig. 8
f_{wadd}	2.0		wing added weight fraction
W_{fixed}	14700	N	fixed weight
$C_{L, \text{max}}$	1.5		max C_L , flaps down
ρ	0.91	kg/m^3	air density, 3000m
ρ_{sl}	1.23	kg/m^3	air density, sea level
μ	1.69×10^{-5}	kg/(sm)	dynamic viscosity, 3000m
e	0.95		wing spanwise efficiency
A_{prop}	0.785	m^2	propeller disk area
η_v	0.85		propeller viscous efficiency
η_{eng}	0.35		engine efficiency
h_{fuel}	46×10^6	J/kg	fuel heating value

E. Vector Variables

Because we need to analyze the vehicle in three different flight conditions (outbound, return, sprint), the following decision variables are 3-vectors instead of scalars:

$$V, C_L, C_D, C_{D\text{fuse}}, C_{Dp}, C_{Di}, T, W, \text{Re}, \eta_i, \eta_{\text{prop}}, \eta_0$$

When any of these variables appears in a constraint, that constraint is enforced for each element of the vector.

F. GP Formulation of Example Design Problem

The design problem is given by the following GP:

$$\text{minimize } W_{\text{fuel, out}} + W_{\text{fuel, ret}}$$

subject to

Steady Level Flight Relations

$$W = \frac{1}{2} \rho V^2 C_L S$$

$$T \geq \frac{1}{2} \rho V^2 C_D S$$

$$\text{Re} = \frac{\rho V S^{1/2}}{A^{1/2} \mu}$$

Landing Flight Condition

$$W_{\text{MTO}} \leq \frac{1}{2} \rho_{\text{sl}} V_{\text{stall}}^2 C_{L, \text{max}} S$$

$$V_{\text{stall}} \leq 38$$

Sprint Flight Condition

$$P_{\text{max}} \geq \frac{T_{\text{sprint}} V_{\text{sprint}}}{\eta_{0, \text{sprint}}}$$

$$V_{\text{sprint}} \geq 150$$

Drag Model

$$C_D \geq \frac{0.05}{S} + C_{Dp} + \frac{C_L^2}{\pi e A}$$

$$1 \geq \frac{2.56C_L^{5.88}}{\text{Re}^{1.54}\tau^{3.32}C_{Dp}^{2.62}} + \frac{3.8 \times 10^{-9}\tau^{6.23}}{C_L^{92}\text{Re}^{1.38}C_{Dp}^{9.57}} + 0.0022 \frac{\text{Re}^{0.14}\tau^{0.033}}{C_L^{0.01}C_{Dp}^{0.73}} + \dots$$

$$\dots 1.19 \times 10^4 \frac{C_L^{9.78}\tau^{1.76}}{\text{Re}C_{Dp}^{0.91}} + \frac{6.14 \times 10^{-6}C_L^{6.53}}{\text{Re}^{0.99}\tau^{0.52}C_{Dp}^{5.19}}$$

Propulsive Efficiency

$$\eta_0 \leq \eta_{\text{eng}}\eta_{\text{prop}}$$

$$\eta_{\text{prop}} \leq \eta_i\eta_v$$

$$4\eta_i + \frac{T\eta_i^2}{\frac{1}{2}\rho V^2 A_{\text{prop}}} \leq 4$$

Range Constraints

$$R \geq 5000 \times 10^3$$

$$z_{\text{bre}} \geq \frac{gRT}{h_{\text{fuel}}\eta_0 W}$$

$$\frac{W_{\text{fuel}}}{W} \geq z_{\text{bre}} + \frac{z_{\text{bre}}^2}{2} + \frac{z_{\text{bre}}^3}{6} + \frac{z_{\text{bre}}^4}{24}$$

Weight relations

$$W_{\text{pay}} > 500g$$

$$\tilde{W} \geq W_{\text{fixed}} + W_{\text{pay}} + W_{\text{eng}}$$

$$W_{\text{zfw}} \geq \tilde{W} + W_{\text{wing}}$$

$$W_{\text{eng}} \geq 0.0372P_{\text{max}}^{0.803}$$

$$\frac{W_{\text{wing}}}{f_{\text{wadd}}} \geq W_{\text{web}} + W_{\text{cap}}$$

$$W_{\text{outbound}} \geq W_{\text{zfw}} + W_{\text{fuel,ret}}$$

$$W_{\text{MTO}} \geq W_{\text{outbound}} + W_{\text{fuel,out}}$$

$$W_{\text{sprint}} = W_{\text{outbound}}$$

Wing Structural Models

$$2q \geq 1 + p$$

$$p \geq 1.9$$

$$\tau \leq 0.15$$

$$\bar{M}_r \geq \frac{\tilde{W}Ap}{24}$$

$$0.92\bar{w}\tau\bar{t}_{\text{cap}}^2 + \bar{I}_{\text{cap}} \leq \frac{0.92^2}{2}\bar{w}\tau^2\bar{t}_{\text{cap}}$$

$$8 \geq \frac{N_{\text{lift}}\bar{M}_r A q^2 \tau}{S\bar{I}_{\text{cap}}\sigma_{\text{max}}}$$

$$12 \geq \frac{A\tilde{W}N_{\text{lift}}q^2}{\tau S\bar{I}_{\text{web}}\sigma_{\text{max, shear}}}$$

$$v^{3.94} \geq 0.86p^{-2.38} + 0.14p^{0.56}$$

$$W_{\text{cap}} \geq \frac{8\rho_{\text{cap}}g\bar{w}\bar{t}_{\text{cap}}S^{3/2}v}{3A^{1/2}}$$

$$W_{\text{web}} \geq \frac{8\rho_{\text{web}}g r_h \tau \bar{I}_{\text{web}} S^{3/2} v}{3A^{1/2}}$$

	outbound	return	sprint
V	69.2	65.78	150
C_L	0.5523	0.5521	0.1175
C_D	0.01292	0.01293	0.007883
$C_{D\text{fuse}}$	0.001725	0.001725	0.001725
C_{Dp}	0.005546	0.005559	0.005903
C_{Di}	0.005647	0.005644	0.0002558
T	816	737.8	2340
W	3.489e+04	3.151e+04	3.489e+04
Re	4.716e+06	4.483e+06	1.022e+07
η_i	0.9028	0.9027	0.9362
η_{prop}	0.7674	0.7673	0.7958
η_0	0.2686	0.2686	0.2785

Table 3. Globally-optimal values for flight-condition-dependent decision variables in the example problem.

	outbound	return
W_{fuel}	3731	3374
z_{bre}	0.1016	0.1017

Table 4. Globally-optimal values for fuel-related outbound/return variables in the example problem.

A	18.1
\bar{I}_{cap}	1.908e-05
\bar{M}_r/c_r	3.231e+04
P_{max}	1.26e+06
R	5e+06
S	28.99
V_{stall}	38
v	0.786
p	1.9
q	1.45
τ	0.15
t_{cap}	0.004273
t_{web}	0.0005907
W_{cap}	4347
W_{zfw}	3.151e+04
W_{eng}	2949
W_{mto}	3.862e+04
W_{pay}	4900
\tilde{W}	2.255e+04
W_{web}	135.2
W_{wing}	8965

Table 5. Globally-optimal values for scalar decision variables in the example problem.

G. Optimal Solution of Example Design Problem

With the GP defined, we turn to a commercially-available solver, MOSEK, which finds a global optimum for all unknowns in less than 0.01 seconds on a standard desktop computer (Tables 3-5).

VII. Current Limitations and Future Perspectives

What if we encounter a model that is not GP-compatible?

GP-based design is by no means universally applicable. It only applies to 1) analytical models that can be expressed in terms of posynomials, or 2) data that is well-approximated by log-convex functions. While this article argues that these conditions exist surprisingly often, it is clear that many relationships cannot be captured directly in a standard GPⁱ.

Discrete decisions such as number of powerplants, elevator vs. canard, material choice, etc., can be modeled by *integer* variables. Including these configuration choices in the design optimization results in a feasible set (design space) that is not convex. Such a problem can be modeled as a *mixed-integer GP*. With the exception of small problems, mixed-integer optimization problems are significantly more difficult than convex programs. Most solution methods make some sacrifices, such as no longer guaranteeing global optimality, in return for finding an acceptable solution in a reasonable amount of time. Solution methods for mixed-integer GPs are an active area of research [25, 33].

Quasi-convex functions are those functions for which every *level set* is a convex set [23]. Informally, quasi-convexity extends the concept of unimodality to higher dimensional functions. When a model or physical relationship is described by a quasi-convex (or log-quasi-convex) function, it *may* be possible to find a nonlinear change of variables under which the relevant functions become convex (or log-convex). An example is the change of variables (60) used in Section V to model wing structural properties.

There are many techniques available for handling more general (not necessarily log-convex) models within the GP framework. All such methods sacrifice guarantees of global optimality, and are therefore qualitatively similar to solving general nonlinear programs. They differ from NLP methods, however, in their treatment of large subsets of the problem in a convex form. Notable methods of this type include *signomial programming* and the *convex-concave procedure* [23].

How does the GP approach relate to Multidisciplinary Design Optimization architectures?

In their 1993 paper, Cramer et al. outlined standard formulations for MDO [2]. These include all-at-once (AAO, sometimes called SAND), individual discipline feasible, and multidisciplinary feasible. The key differences among the approaches are degree to which optimization is centralized or decentralized, and what kind of feasibility is maintained during each optimization iteration. The GP formulation is an AAO approach, characterized by a centralized solver and lack of disciplinary separation.

Historically, AAO approaches have performed well in benchmarking tests against other MDO algorithms [45]. Some believe they are the most computationally tractable of all MDO approaches [2], but they are sometimes written off because they create large problems with many equality constraints. We hold hope for the GP version of AAO, because the restriction to convex constraint sets enables efficient solutions that scale to problems with hundreds of thousands of constraints [33].

Like other AAO approaches, GP centralizes the optimization task, thereby largely eliminating organizational and communication challenges. Coordination problems are limited to agreeing on a common modeling language (the variable names, notionally). When designers want to improve a model, capture a new effect, or model a new tradeoff, they simply add or update the corresponding constraint.

Can black-box analysis routines be called by a GP solver?

No. GP solvers accept as input a standard parameterization (5) for the GP to be solved. There is no way for a GP solver to directly interface with an arbitrary analysis routine. Such an interface would void all guarantees of global optimality and efficient optimization provided by GP.

As described in Section V, however, one option is to sample disciplinary solvers offline, and fit GP-compatible surrogate models to the

resulting data. This approach is possible when the data set, or relevant subsets thereof, are well-approximated by log-convex functions. Because high-dimensional design spaces would require an impractical amount of data, model order reduction techniques are relevant here.

Does this approach maintain ‘feasibility’ at every iteration?

In the context of MDO, *feasibility* can have two meanings. One notion of feasibility refers to an equilibrium condition where the inputs and outputs of the various analysis equations agree with equality. Some formulations impose constraints that drive the residual of these quantities to zero. MDO formulations vary as to when they enforce these feasibility equality constraints.

In geometric programming, and more generally in convex optimization, *feasibility* has the second and more common meaning: it is a condition where all equality and inequality constraints are satisfied. In this paper, we used the posynomial equality relaxation described in Section B to expand the feasible set of the optimization problem.

Interior point methods for solving GPs stay inside this expanded feasible set during optimization. For the analysis models presented herein, equality will hold at the final optimum.

What is the objective function? Does this approach support multi-objective optimization?

The GP approach gives complete flexibility with regard to the objective function. Concretely, the objective may be any monomial or posynomial function of the decision variables. In practice, one would choose to maximize (or minimize) some key criteria of interest or combination thereof.

In multi-objective optimization, we are interested in sweeping out a Pareto frontier corresponding to the tradeoff surface among a number of variables. There are two ways to sweep out such a Pareto frontier within the GP approach:

1. We can formulate a posynomial objective function corresponding to a weighted combination of some criteria of interest (a weighted combination of range and payload, for example). We then sweep the weights through a convex set, solving the GP at each point.
2. We can pick one variable as the objective (range, for example), and constrain other variables of interest (payload, for example) to be greater than (or less than) some value s . We then sweep s over a range of interest, solving the GP at each point.

In both cases, the speed of GP solution methods allows each point to be calculated extremely quickly, freeing up the decision maker to consider a larger number of possible Pareto-optimal designs.

Acknowledgments

We thank Karen Willcox for many insightful conversations and comments. We also thank two anonymous reviewers for their thorough and helpful suggestions. This work was supported by an NSF Graduate Research Fellowship.

References

- [1] J. Sobieszczanski-Sobieski, R. H., “Multidisciplinary aerospace design optimization: survey of recent developments,” *Structural Optimization*, Vol. 14, 1997, pp. 1–23.
- [2] Cramer, E. J., Dennis, Jr, J., Frank, P. D., Lewis, R. M., and Shubin, G. R., “Problem formulation for multidisciplinary optimization,” *SIAM Journal on Optimization*, Vol. 4, No. 4, 1994, pp. 754–776.
- [3] Martins, J. and Lambe, A. B., “Multidisciplinary design optimization: Survey of architectures,” *AIAA Journal*, 2012.
- [4] Alexandrov, N. M. and Lewis, R. M., “Analytical and computational aspects of collaborative optimization for multidisciplinary design,” *AIAA Journal*, Vol. 40, No. 2, 2002, pp. 301–309.
- [5] Kim, H. M., Michelena, N. F., Papalambros, P. Y., and Jiang, T., “Target cascading in optimal system design,” *Journal of Mechanical Design*, Vol. 125, 2003, pp. 474.

ⁱOne of our hopes is that the *process* of classifying models according to convexity can help inform more efficient optimization architectures.

- [6] Kroo, I. M., "MDO for Large-Scale Design," *Multidisciplinary Design Optimization: State-of-the-Art*, edited by N. Alexandrov and M. Y. Husaini, SIAM, 1997, pp. 22–44.
- [7] Kodiyalam, S. and Sobieszcanski-Sobieski, J., "Bilevel integrated system synthesis with response surfaces," *AIAA Journal*, Vol. 38, No. 8, 2000, pp. 1479–1485.
- [8] Sobieszcanski-Sobieski, J., Altus, T. D., Phillips, M., and Sandusky, R., "Bilevel integrated system synthesis for concurrent and distributed processing," *AIAA Journal*, Vol. 41, No. 10, 2003, pp. 1996–2003.
- [9] Sobieski, I. and Kroo, I., "Collaborative optimization using response surface estimation," *AIAA journal*, Vol. 38, No. 10, 2000, pp. 1931–1938.
- [10] Martin, J. D. and Simpson, T. W., "Use of kriging models to approximate deterministic computer models," *AIAA journal*, Vol. 43, No. 4, 2005, pp. 853–863.
- [11] Bui-Thanh, T., Willcox, K., and Ghattas, O., "Model Reduction for Large-Scale Systems with High-Dimensional Parametric Input Space," *SIAM Journal on Scientific Computing*, 2008.
- [12] Willcox, K. and Peraire, J., "Balanced model reduction via the proper orthogonal decomposition," *AIAA Journal*, Vol. 40, No. 11, 2002, pp. 2323–2330.
- [13] Pironneau, O., "On optimum profiles in Stokes flow," *Journal of Fluid Mechanics*, Vol. 59, No. 01, 1973, pp. 117–128.
- [14] Pironneau, O., "On optimum design in fluid mechanics," *Journal of Fluid Mechanics*, Vol. 64, No. 01, 1974, pp. 97–110.
- [15] Jameson, A., "Aerodynamic design via control theory," *Journal of Scientific Computing*, Vol. 3, No. 3, 1988, pp. 233–260.
- [16] Jameson, A., "Computational aerodynamics for aircraft design," *Science(Washington)*, Vol. 245, No. 4916, 1989, pp. 361–371.
- [17] Jameson, A., Martinelli, L., and Pierce, N., "Optimum aerodynamic design using the Navier–Stokes equations," *Theoretical and Computational Fluid Dynamics*, Vol. 10, No. 1–4, 1998, pp. 213–237.
- [18] Baysal, O. and Eleshaky, M. E., "Aerodynamic design optimization using sensitivity analysis and computational fluid dynamics," *AIAA journal*, Vol. 30, No. 3, 1992, pp. 718–725.
- [19] Elliott, J. and Peraire, J., "Practical three-dimensional aerodynamic design and optimization using unstructured meshes," *AIAA journal*, Vol. 35, No. 9, 1997, pp. 1479–1485.
- [20] Martins, J. R., Alonso, J. J., and Reuther, J. J., "A coupled-adjoint sensitivity analysis method for high-fidelity aero-structural design," *Optimization and Engineering*, Vol. 6, No. 1, 2005, pp. 33–62.
- [21] RA Martins, J. R., Alonso, J. J., and Reuther, J. J., "High-fidelity aerostructural design optimization of a supersonic business jet," *Journal of Aircraft*, Vol. 41, No. 3, 2004, pp. 523–530.
- [22] Robinson, T., Eldred, M., Willcox, K., and Haimes, R., "Surrogate-based optimization using multifidelity models with variable parameterization and corrected space mapping," *Aiaa Journal*, Vol. 46, No. 11, 2008, pp. 2814–2822.
- [23] Boyd, S. and Vandenberghe, L., *Convex Optimization*, Cambridge University Press, New York, NY, USA, 2004.
- [24] Boyd, S. P., Kim, S.-J., Patil, D. D., and Horowitz, M. A., "Digital Circuit Optimization via Geometric Programming," *Operations Research*, Vol. 53, 2005, pp. 899–932.
- [25] Chiang, M., "Geometric programming for communication systems," *Commun. Inf. Theory*, Vol. 2, July 2005, pp. 1–154.
- [26] Babakhani, A., Lavaei, J., Doyle, J., and Hajimiri, A., "Finding Globally Optimum Solutions in Antenna Optimization Problems," *IEEE International Symposium on Antennas and Propagation*, 2010.
- [27] Duffin, R. J., Peterson, E. L., and Zener, C., *Geometric programming: theory and application*, Wiley New York, 1967.
- [28] Morris, A., "Structural optimization by geometric programming," *International Journal of Solids and Structures*, Vol. 8, No. 7, 1972, pp. 847–864.
- [29] Peterson, E. L., "Geometric programming," *Advances in Geometric Programming*, Springer, 1980, pp. 31–94.
- [30] Beightler, C. S. and Phillips, D. T., *Applied geometric programming*, Vol. 150, Wiley New York, 1976.
- [31] Wilde, D., *Globally optimal design*, Wiley interscience publication, Wiley, 1978.
- [32] Nesterov, Y. and Nemirovsky, A., "Interior-point polynomial methods in convex programming, volume 13 of Studies in Applied Mathematics," *SIAM, Philadelphia, PA*, 1994.
- [33] Boyd, S., Kim, S.-J., Vandenberghe, L., and Hassibi, A., "A tutorial on geometric programming," *Optimization and Engineering*, 2007.
- [34] Mehrotra, S., "On the implementation of a primal-dual interior point method," *SIAM Journal on optimization*, Vol. 2, No. 4, 1992, pp. 575–601.
- [35] Alonso, J. J., "AA222: Introduction to Multidisciplinary Design Optimization," Lecture Notes.
- [36] Mosek-APs, "MOSEK version 6.0.0.148," Free academic license available at <http://mosek.com/resources/academic-license/personal-license/>.
- [37] Drela, M., "Course Notes," MIT Unified Engineering.
- [38] Hoburg, W. and Abbeel, P., "Fitting Geometric Programming Models to Data," *In review*, 2014.
- [39] Drela, M., "QPROP Formulation - Theory Document," MIT Aero & Astro.
- [40] Anderson, J. D., *Fundamentals of Aerodynamics*, McGraw-Hill, 3rd ed., 2001.
- [41] Drela, M., "XFOIL Subsonic Airfoil Development System," Open source software available at <http://web.mit.edu/drela/Public/web/xfoil/>.
- [42] Marquardt, D. W., "An Algorithm for Least-Squares Estimation of Non-linear Parameters," *Journal of the Society for Industrial and Applied Mathematics*, Vol. 11, No. 2, 1963, pp. 431–441.
- [43] Drela, M., "TASOPT 2.00 – Transport Aircraft System OPTimization," *MIT N+3 Final Report*, March 2010.
- [44] Raymer, D. P. et al., *Aircraft design: a conceptual approach*, American Institute of Aeronautics and Astronautics Reston, 2006.
- [45] Tedford, N. P. and Martins, J. R., "Benchmarking multidisciplinary design optimization algorithms," *Optim Eng*, February 2009.