Efficient and Reliable Aircraft Multidisciplinary Design Optimization via Signomial Programming

Martin A. York[1], Berk Ozturk[2], Edward Burnell[3], and Warren W. Hoburg[4]
Massachusetts Institute of Technology, Cambridge, MA, 02139

This paper presents benefits of using signomial programming for aircraft multidisciplinary design optimization (MDO). Building from prior work, a full aircraft signomial programming MDO tool was developed and is shown to have a number of advantages relative to existing tools, most notably increased solution speed and the ability to compute parameter and constraint sensitivities relative to the objective at virtually no computational cost. The signomial programming tool is 17 times faster for a medium scale benchmark problem and 71 times faster for a large scale benchmark problem than Transport Aircraft System Optimization (TASOPT), a comparable aircraft MDO tool. Additionally, the signomial programming tool is shown to reliably converge for a variety of objective functions and evaluate non-traditional aircraft configurations which include boundary layer ingestion and double bubble fuselages.

Nomenclature

AR_w = wing aspect ratio
b_w = wing span
D = drag
$\frac{L}{D}$ = aircraft lift to drag ratio
F = thrust
M_{min} = minimum cruise Mach number
t_{total} = total mission flight time
V_{ne} = never exceed speed
W_{empty} = aircraft empty weight
W_{engine} = engine weight
W_{lg} = landing gear weight
$W_{f_{\text{total}}}$ = total fuel weight
$(\cdot)_i$ = flight segment i quantity

I. Introduction

A. Background

Physics-based models, rather than regression models, are necessary when exploring new aircraft concepts. Furthermore, system-level optimization is needed to determine the multidisciplinary optimum of the airframe, engine, and operations design space[1]. A number of physics based, system level, multidisciplinary design optimization (MDO) tools have been developed and are currently in use. However, there is a need for new architectures that exhibit fast convergence for medium and large scale problems[2].

---

[1] Graduate Student, Department of Aeronautics and Astronautics; currently Second Lieutenant, United States Air Force.
[2] Graduate Student, Department of Aeronautics and Astronautics.
[3] Graduate Student, Department of Mechanical Engineering.
[4] Assistant Professor, Department of Aeronautics and Astronautics; currently Astronaut Candidate, NASA. AIAA Member.

One technique for improving computational efficiency is to solve particular forms of optimization problems rather than general non-linear programs. Hoburg et al.[3] successfully formulates a basic aircraft design problem as a geometric program (GP), which is a type of convex optimization problem. GPs with thousands of design variables can be solved on a personal laptop in a number of seconds. GPs guarantee convergence to a global optimum, and allow for parameter and constraint sensitivities to be computed at essentially zero computational cost[4]. A caveat of GPs is all equations must be posed in a manner compatible with GPs. While a number of relevant relationships fit the required form, it can prove limiting. This motivates the use of signomial programming. See Appendix A for a detailed description of geometric programming.

Signomial programs (SPs), which are difference-of-convex programs, are non-convex extensions of GPs. A detailed description of signomial programming is available in Appendix B. Restrictions on the form of SP constraints are less stringent than the restrictions on GP constraints. SPs have many of the advantages of GPs, such as their relative speed compared to general non-linear problems and low cost computation of optimal sensitivities. However, unlike GPs, SPs provide no guarantee of global optimality. Kirschen et al. [5] develops physics based, signomial programing compatible models for aircraft wings, fuselages, horizontal tails, vertical tails and landing gear. York et al.[6] develops a full 1D core+fan flow path simulation turbofan engine model compatible with signomial programming.

B. Present Work

In this paper, the subsystem models developed in [5, 6] have been combined into a single SP compatible aircraft conceptual design MDO tool of comparable fidelity to Transport Aircraft System Optimization (TASOPT) [1]. The SP design tool is used to perform a series of case studies which illustrate advantages of signomial programming. In Section III A, the SP design tool is shown to perform a single mission aircraft optimization 17 times faster than TASOPT and a multi-mission aircraft optimization 71 times faster than TASOPT. Section III B presents a sample sensitivity analysis and Section III C demonstrates the SP tool's ability to reliably change objective functions. Finally, the SP tool's ability to model different aircraft configurations is illustrated in Section III D where results for 737, 777, and D8.2[7] aircraft models are presented.

II. Model Overview

The full aircraft system model is a set of coupled subsystem models, leveraging the models developed by Kirschen et al. [5] and York et al[6]. The model was constructed with GPkit [8] using its built-in framework for multi-point optimization. Descriptions of GPkit and the multi-point optimization formulation as well as a qualitative overview of the subsystem models are provided below. The code used in this model is publicly available at `https://github.com/hoburg/SPaircraft`.

A. Optimization Formulation

All SPs were formulated on a laptop computer using GPkit [8], and solved using a commercially-available solver, MOSEK [9]. GPkit, developed at MIT, is a Python package that enables the fast and intuitive formulation of GPs and SPs. GPkit has a built-in heuristic for solving SPs via a series of GP approximations, and binds with open source and commercial interior point solvers to solve the individual GPs. The presented models are solved with the relaxed constants heuristic detailed in [5].

B. Multi-Point Optimization Formulation

Each individual GP is a non-hierarchical collection of every constraint (or its local approximation) in the model. However, during model development, a hierarchy of models such as that in Figure 1 helps ensure the appropriate constraining connections between subsystems are made. Many different modular decompositions of the full-system model are possible, but the combination of two design rules generally leads to a single obvious and highly reusable decomposition.

The first rule is a strict maintenance of hierarchy: models can reference only the variables of models that are at the same or lower level of hierarchy than themselves. This rule is useful in the design of any component-decomposed system such as the SP aircraft model and helps determine when a constraint that seems like part of a low-level subsystem is best considered in a higher-level system.

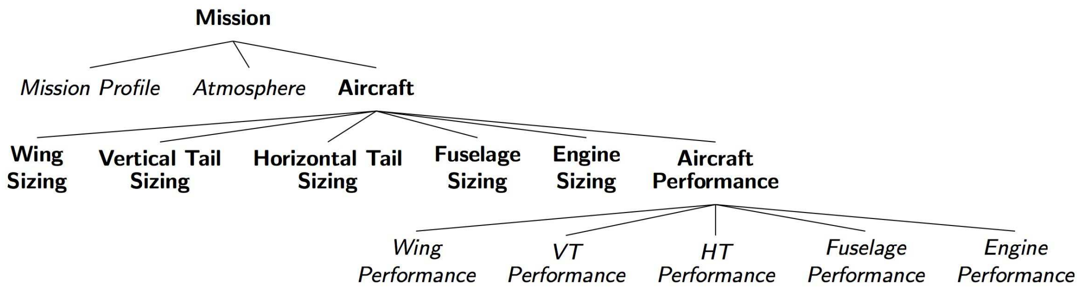


Fig. 1 Hierarchy of the presented aircraft model. Models that include sizing variables are bolded while models that include performance variables are italicized. There are models that contain both kinds of variables.

The second rule is a separation of "sizing" and "performance" variables into separate models. Sizing models contain all variables and constraints that do not change between operating points, such as component weights and dimensions. Performance models contain all constraints and variables that change between operating points, such as air speeds, lift coefficients, and fuel quantities. Sizing models contain a pointer to their companion performance model. This model separation allows the modeler to specify a scalar performance model but create it in a 'vectorized' environment which extends each variable in the original scalar performance model across the vector of operating points. For example, the constraint thrust is greater than or equal to drag could be written in a scalar performance model as $F \geq D$ and then be vectorized across N operating points. After vectorization, the original constraint $F \geq D$ becomes the N unique constraints below.

$$\begin{gathered} F_1 \geq D_1 \\ F_2 \geq D_2 \\ \vdots \\ F_N \geq D_N \end{gathered}$$

Figure 2 provides a visual representation of sizing and performance models. The technique is not restricted to aerospace applications and can be used for any multi-point optimization. Vectorization allows the airplane design problem to be extended to a fleet design problem with a single line of code. Together, these two model development rules help specify simple submodels, making it easier for modelers to collaborate and use models written by others.

C. Aircraft Subsystem Models

Through the model hierarchy detailed in Figure 1, the subsystem models described below are linked through shared variables. For example, the wing structural model depends on engine weight and the fuselage structural model depends on maximum tail aerodynamic loads. A full description of subsystem variable linking can be found in [5].

1. Fuselage

The fuselage model is adapted from the model in [5], and borrows heavily from TASOPT [1]. Modifications, described in Appendix IV C, were made to support double bubble fuselages in addition to traditional tube fuselages. Fuselage sizing constraints include pressure loads, y-axis and z-axis bending moments, and floor loads.

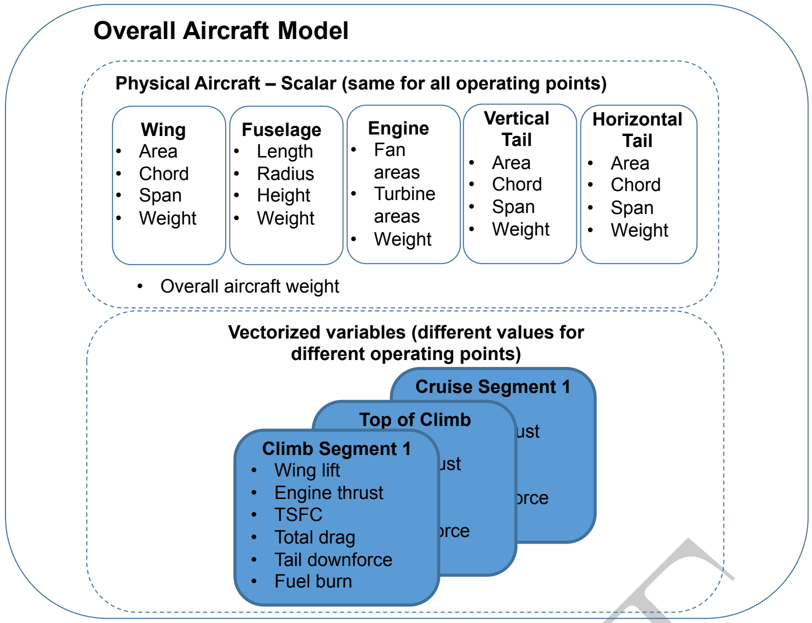


Fig. 2 Aircraft model architecture.

2. Engine

The SP model utilizes the full 1D core+fan flow path simulation turbofan engine model developed by York et al. [6]. An additional boundary layer ingestion model used for the SP D8.2 is described in Appendix D. The nacelle drag model in [1] is adopted with one modification: the nacelle skin friction coefficient is assumed to be that of a turbulent flat plate.

3. Wing

The wing model is taken directly from [5]. The model includes a physics based structural model, geometry based induced drag and lift curve slope estimation, drag fits to modern transonic airfoils, and a fuel tank volume model.

4. Vertical and Horizontal Tails

This work leverages the tail models in [5]. The vertical tail is sized for both takeoff engine out and a minimum required yaw acceleration rate at flare.

The horizontal tail is sized to provide a minimum static margin at the forward and aft CG locations (this work includes the CG model in [5]). The SP model can implement both pi-tails and conventional tails.

The vertical tail surfaces utilize the same structural model as the wing. To facilitate modeling pi-tails, the horizontal tail has a unique structural model described in Appendix E.

5. Landing Gear

The landing gear model is taken directly from [5]. The model includes aircraft geometry constraints as well as taxi and landing load cases.

6. Mission Profile

The mission profile is described in Appendix F. It includes climb and cruise segments, both of which can be discretized into an arbitrary number of sub-segments. For the purposes of this paper, three climb and two cruise segments were used. Climb performance is computed using an excess power formulation. For each cruise segment, the optimizer can either fly level or execute a cruise climb (cruise climb rate, cruise altitude, and lift coefficient are optimized).

Table 1 Comparison of SP and TASOPT solution times for different 737-800 models. The SP model experiences a 1.5 times slow down when moving from the single to multi-mission solve while TASOPT experiences a 6.3 times slow down.

Model	SP Solve Time	TASOPT Solve Time	Number of Variables in SP Model
Pure Analysis	N/A	< 1 sec	N/A
Single Point Optimization	7.29 sec	2 min 4 sec	2,002
Two Mission Optimization	10.9 sec	12 min 58 sec	3,475

7. Atmosphere

The atmosphere model is taken directly from [5].

III. Case Studies

A. Solution Time Comparison

Solution times for the SP model and TASOPT are presented in Table 1. TASOPT uses a traditional gradient based optimization method[1]. The two-mission optimization consists of optimizing a single aircraft to fly both a 3,000 nm and a 2,000 nm mile mission, each with the same payload as the 737 in Section III D.

The SP model solves 17 times faster than TASOPT for the single mission case and 71 times faster than TASOPT for the multi-mission case. The SP model experiences a 1.5 times slow down when moving from the single to multi-mission solve while TASOPT experiences a 6.3 times slow down. This suggests the SP formulation scales better to large problems than traditional gradient-based optimization formulations. There are modeling differences between the SP model and TASOPT which may affect the timing results. However, these modeling differences are minor, and do not substantially skew the final solution or the time comparison [1, 5, 6].

B. Sensitivity Analysis

The SP model automatically computes the sensitivity of each constant and constraint with respect to the optimal objective value. Equation 1 defines parameter sensitivity and Equation 2 defines constraint sensitivity [10]. Appendix G provides additional technical information regarding sensitivities.

$$\text{Parameter Sensitivity} = \frac{\text{Fractional Objective-Function Change}}{\text{Fractional Parameter Change}} \tag{1}$$

$$\text{Constraint Sensitivity} = \frac{\text{Fractional Objective-Function Change}}{\text{Fractional Change In Constraint Tightness}} \tag{2}$$

These sensitivities are useful in engineering design for two reasons. The first is to determine which areas of a physical design should be improved. For example, if the sensitivity to burner pressure drop is very large, it is advantageous to focus effort on reducing the burner pressure drop. Sensitivities are also a useful guide for model development. If the sensitivity to a parameter is high, then it is important to either know the value of that parameter with a high degree of certainty or replace the parameter with a more detailed model. However, if the sensitivity of a parameter is low, uncertainty in the value of the parameter is unlikely to have a large effect on the model's solution. Table 2 presents selected sensitivity information for the optimal 737 model presented in Section III D. Table 3 presents the same parameter sensitivities for the D8.2 model discussed in Section III D.

To demonstrate the accuracy of the computed sensitivities, the 737 model was optimized for minimum fuel burn for mission ranges of 3,000 nm and 2,995 nm. The 3,000 nm fuel burn was 41,914 pounds while the 2,995 nm fuel burn was 41,826 pounds, 0.21% less. The percent change in fuel burn divide by the percent change in range is equal to 1.271. The sensitivity to mission range is 1.266.

Table 2 Selected sensitivities for the optimal 737 presented in Section III D

Parameter	Sensitivity
Avg. Passenger Weight (incl. payload)	0.83
Wing Max Allowed Tensile Stress	-0.30
Range	1.3
V_{ne}	0.32
Reserve Fuel Fraction	0.26
M_{min}	0.43
Max Skin Stress	-0.036
Burner Efficiency	-0.51

Table 3 Selected sensitivities for the optimal D8.2 presented in Section III D.

Parameter	Sensitivity
Avg. Passenger Weight (incl. payload)	0.66
Wing Max Allowed Tensile Stress	-0.23
Range	1.1
V_{ne}	0.23
Reserve Fuel Fraction	0.21
M_{min}	0.62
Max Fuselage Skin Stress	-0.037
Burner Efficiency	-1.2

It is interesting to analyze how sensitivities change as parameters vary. An example is presented in Figure 3. As the max allowed turbine inlet temperature increases, the engine's power density increases and weight decreases. As engine weight becomes a smaller proportion of total aircraft weight, further reductions in engine weight have decreasing returns with respect to overall system performance. Hence, the sensitivity to engine system weight decreases as max allowed turbine inlet temperature increases.

C. Model Robustness Across Objective Functions

Geometric and signomial programs are bags of constraints that are solved all at once via an interior point method. These methods satisfy stringent convergence criteria even with the naive initial guess of one for all variables. In the SP model there are no parameter tuning or weight convergence loops. These factors, along with the mathematically favorable structure of SPs, allow the model solve reliably and efficiently across a variety of objective functions. Table 4 presents key design variables obtained when solving the optimal 737 model for a variety of objective functions. Table 4 does not present an exhaustive set of objectives for which the model converges. The SP model can be solved for any weighted sum of objective functions in Table 4 and supports net present value models. This capability enables aircraft performance to be assessed from the perspective of multiple stakeholders, such as operators and manufacturers.

Table 4 Key design variables for a 737 class aircraft optimized for a variety of objective functions.

Objective	$W_{f_{\mathrm{total}}}$	W_{empty}	b_w	AR_w	W_{engine}	t_{total}	Initial Cruise $\frac{L}{D}$	W_{lg}
$W_{f_{\mathrm{total}}}$	1	1	1	1	1	1	1	1
W_{empty}	1.4	0.72	0.69	0.52	0.65	0.99	0.79	0.96
b_w	2.2	0.91	0.58	0.27	1.1	0.96	0.57	1.42
AR_w	4.5	1.8	0.73	0.21	2.3	0.95	0.52	2.3
W_{engine}	1.2	0.82	0.92	0.95	0.54	1.0	1.1	0.97
t_{total}	2.8	1.6	1.0	0.61	2.3	0.85	0.75	2.3
Initial Cruise $\frac{D}{L}$	1.5	1.0	1.0	0.91	0.66	1.0	1.2	1.3
W_{lg}	1.4	0.85	0.93	0.73	0.79	1.0	0.98	0.64

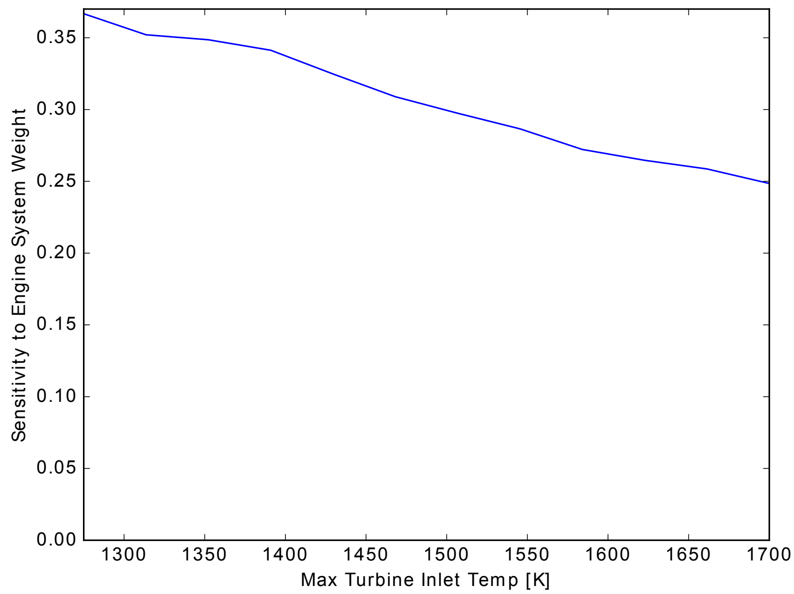


Fig. 3 Sensitivity to engine system weight versus max allowed turbine inlet temperature ($T_{t_{4.1}}$).

Table 5 737-800, D8.2 and 777-300ER model mission parameters.

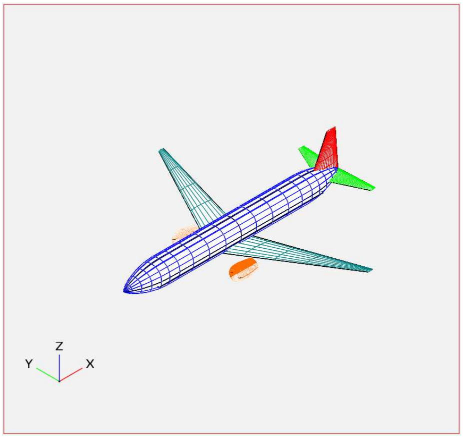
Quantity	737-800	D8.2	777-300ER
Range	3,000 nm	3,000 nm	6,000 nm
Number of Passengers	180	180	450
Minimum Cruise Mach	0.80	0.72	0.84
Payload Weight [lbf]	38,716	38,716	103,541

D. Ability to Model Different Aircraft Configurations

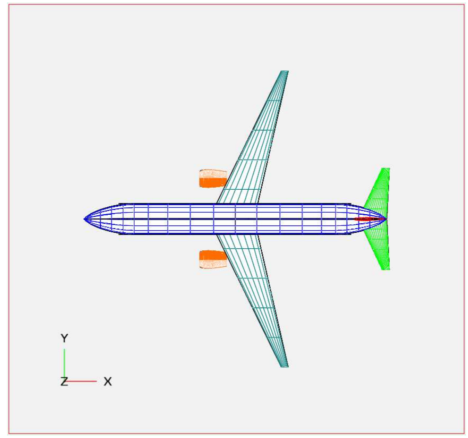
To demonstrate the ability of the SP model to scale and handle different configurations, the SP model was used to optimize three different aircraft architectures : a single aisle airliner similar to a 737-800, a wide body airliner similar to a 777-300ER, and the D8.2[7]. Mission parameters are presented in Table 5. Model results are presented in Table 6. The SP aircraft optimization tool was integrated with OpenVSP[11] to facilitate output visualization. Figures 4 through 6 show selected VSP output for the presented models. In all cases the objective function was total fuel burn. Constant input parameters were selected to match TASOPT input parameters for the example files distributed with TASOPT version 2.16.

Table 6 Results for the SP 737-800, D8.2, and 777-300ER models.

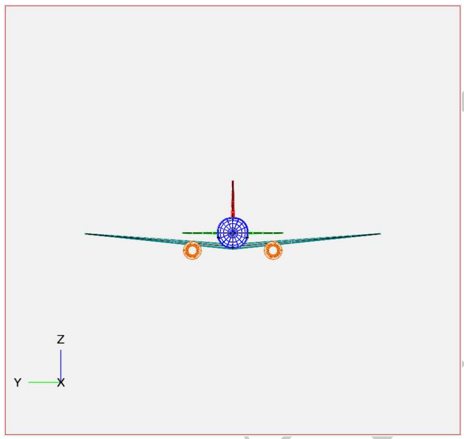
Quantity	737-800 Value	D8.2 Value	777-300ER Value
Takeoff Weight [lbf]	166,317	139,883	593,240
Required Fuel [lbf]	41,914	25,691	208,620
Empty Weight [lbf]	85,702	75,492	281,119
Wing Span [ft]	117.5	140.0	200.0



(a) 737 side view.



(b) 737 top view.



(c) 737 rear view.

Fig. 4 SP 737 VSP outputs.

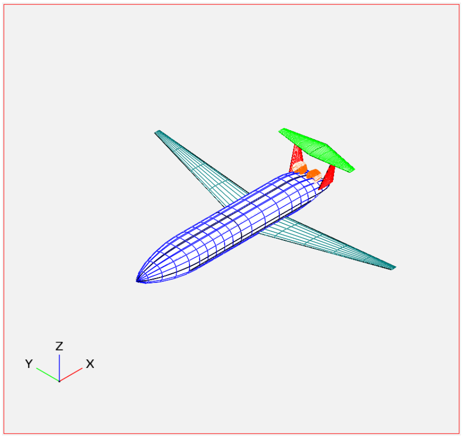
IV. Conclusion

This paper presented the benefits of using signomial programming for aircraft MDO. The signomial programming tool is shown to perform a single mission aircraft optimization 17 times faster and a multi-mission aircraft optimization 71 times faster than TASOPT[1], a comparable existing tool. The ability of signomial programs to automatically compute parameter and constraint sensitivities was demonstrated through a sensitivity analysis. Finally, the signomial programming tool's stability across a range of objective functions and ability to analyze both traditional and non-traditional aircraft configurations was demonstrated. Continued research into aircraft optimization via geometric and signomial programming will likely unearth additional unique capabilities and advantages of these methods.

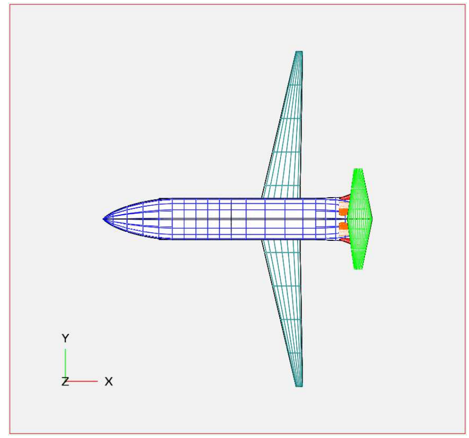
Appendix

A. Geometric Programming

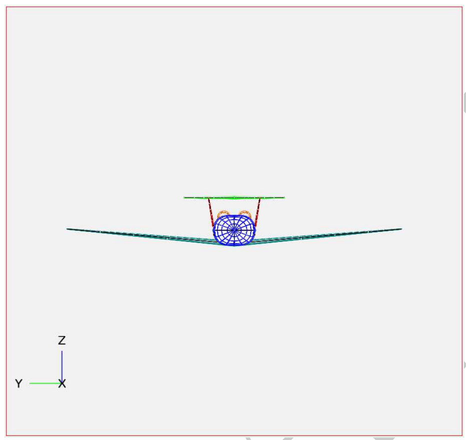
Introduced in 1967 by Duffin et al. [12], a geometric program (GP) is a type of constrained optimization problem that becomes convex after a logarithmic change of variables. Modern interior point methods allow a typical sparse GP with tens of thousands of decision variables and tens of thousands of constraints to be solved in minutes on a desktop computer [13]. These solvers do not require an initial guess, and guarantee convergence to a *global* optimum, assuming a feasible solution exists. If a feasible solution does not exist, the solver will return a certificate of infeasibility. These impressive properties are possible because a GP's objective and constraints consist of only monomial and posynomial functions, which can be transformed into convex functions in log space.



(a) D8.2 side view.



(b) D8.2 top view.



(c) D8.2 rear view.

Fig. 5 SP D8.2 VSP output.

A monomial is a function of the form

$$m(\mathbf{u}) = c\prod_{j=1}^{n} u_j^{a_j} \tag{3}$$

where $a_j \in \mathbb{R}, c \in \mathbb{R}_{++}$ and $u_j \in \mathbb{R}_{++}$. An example of a monomial is the common expression for lift, $\frac{1}{2}\rho V^2 C_L S$. In this case, $\mathbf{u} = (\rho, V, C_L, S)$, $c = 1/2$, and $a = (1, 2, 1, 1)$.

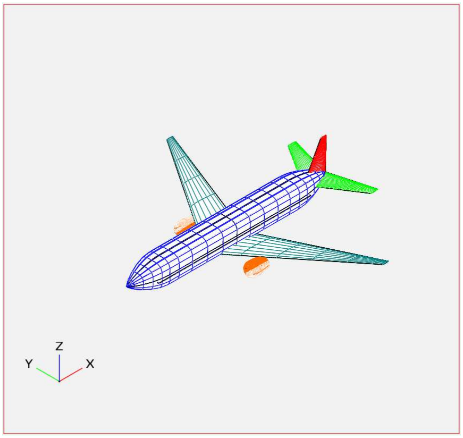
A posynomial is a function of the form

$$p(\mathbf{u}) = \sum_{k=1}^{K} c_k \prod_{j=1}^{n} u_j^{a_{jk}} \tag{4}$$

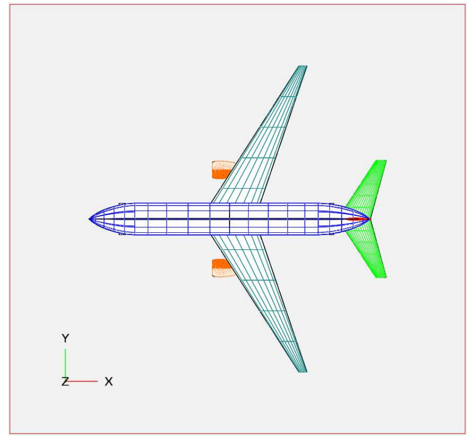
where $a_{jk} \in \mathbb{R}, c_k \in \mathbb{R}_{++}$ and $u_j \in \mathbb{R}_{++}$. A posynomial is a sum of monomials. Therefore, all monomials are also one-term posynomials.

A GP minimizes a posynomial objective function subject to monomial equality and posynomial inequality constraints. A GP written in standard form is

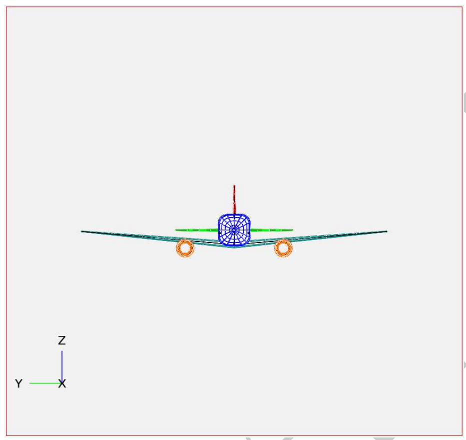
$$\begin{aligned} \text{minimize } & p_0(\mathbf{u}) \\ \text{subject to } & p_i(\mathbf{u}) \leq 1, i = 1, ..., n_p, \\ & m_i(\mathbf{u}) = 1, i = 1, ..., n_m \end{aligned} \tag{5}$$



(a) 777 side view.



(b) 777 top view.



(c) 777 rear view.

Fig. 6 SP 777 VSP output.

where p_i are posynomial functions, m_i are monomial functions, and $\mathbf{u} \in \mathbb{R}^n_{++}$ are the decision variables. Once a problem has been formulated in the standard form (Equation 5), it can be solved efficiently.

B. Signomial Programming

It is not always possible to formulate a design problem as a GP. This motivates the introduction of signomials. Signomials have the same form as posynomials

$$s(\mathbf{u}) = \sum_{k=1}^{K} c_k \prod_{j=1}^{n} u_j^{a_{jk}} \tag{6}$$

but the coefficients, $c_k \in \mathbb{R}$, can now be any (including non-positive) real numbers.

A signomial program (SP) is a generalization of GP where the inequality constraints can be composed of signomial constraints of the form $s(u) \leq 0.$. The log transform of an SP is not a convex optimization problem, but is a difference of convex optimization problem that can be written in log-space as

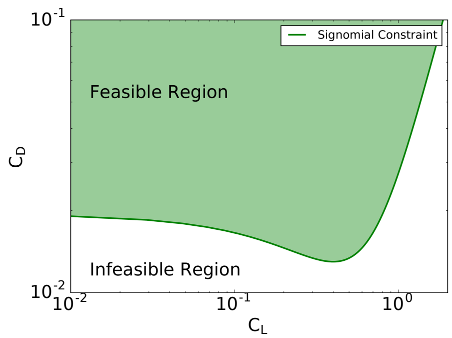
$$\begin{aligned} &\text{minimize } f_0(\mathbf{x}) \\ &\text{subject to } f_i(\mathbf{x}) - g_i(\mathbf{x}) \leq 0, i = 1, ..., m \end{aligned} \tag{7}$$

where f_i and g_i are convex.

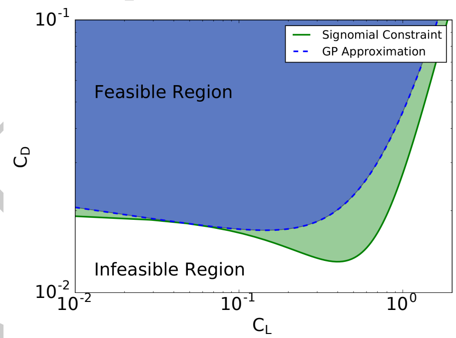
There are multiple algorithms that reliably solve signomial programs to *local* optima [10, 14]. A common solution heuristic, referred to as difference of convex programming or the convex-concave procedure, involves solving a sequence of GPs, where each GP is a local approximation to the SP, until convergence occurs. It is worth noting that the introduction of even a single signomial constraint to any GP turns the GP into a SP, thus losing the guarantee of solution convergence to a global optimum. Despite the possibility of convergence to a local, not global, optimum, SPs are a powerful tool. The convex approximation, $\hat{f}(x)$, to the non-convex signomial in log-space, $f(x) - g(x)$, is constructed such that it always satisfies

$$\hat{f}(x) \geq f(x) - g(x) \quad \forall \quad x \tag{8}$$

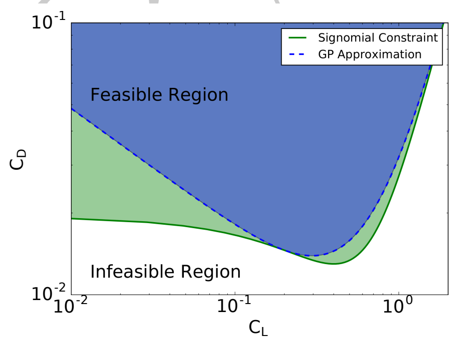
In other words, for each constraint, the feasible set of the convex approximation $\hat{f}(x) \leq 0$ is a subset of the original SP's feasible set, $f(x) - g(x) \leq 0$. This means SP inequalities do not require a trust region, removing the need for trust region parameter tuning and making solving SPs substantially more reliable than solving general nonlinear programs. Figure 7, where a series of convex (GP compatible) constraints approximates a non-convex parabolic drag polar in log space, illustrates this property.



(a) Non-convex signomial inequality drag constraint



(b) Convex approximation about $C_L = 0.05$.



(c) Convex approximation about $C_L = 0.20$.

Fig. 7 A signomial inequality constraint and GP approximations about two different points.

Signomial equality constraints can be approximated by monomials as shown in Figure 8 and may require a trust region. Trust regions were not used in the presented model. Signomial equalities are the least desirable type of constraint due the approximations involved. Most constraints in this work were relaxed to inequalities and checked for tightness by GPkit[8]. For additional details on how signomial equalities are approximated, see Opgenoord et al.[15].

C. Fuselage Modifications

Modifications to the fuselage model in [5] were made to support double bubble fuselages in addition to conventional fuselages.

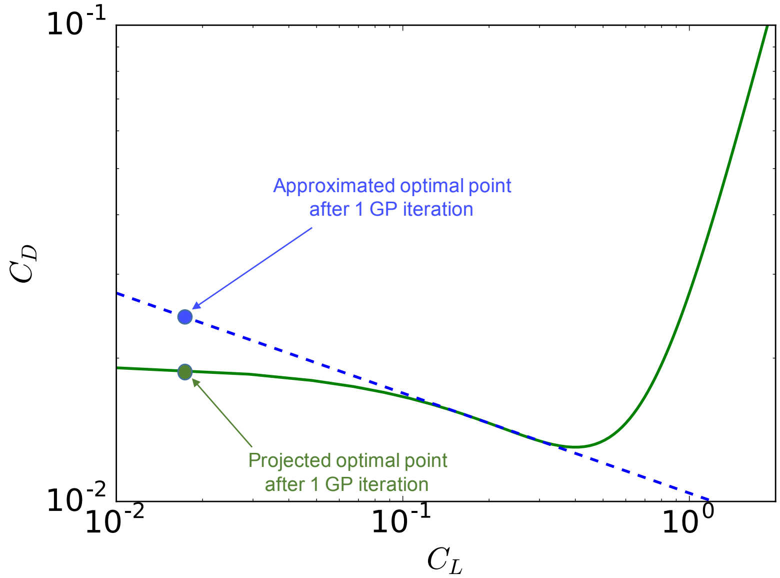


Fig. 8 The signomial equality constraint $C_D = f(C_L)$ and its approximation.

1. Fuselage Terminology

A_{db} = web x-sectional area
A_{fuse} = fuselage x-sectional area
$I_{\mathrm{h_{shell}}}$ = shell horizontal bending inertia
$I_{v_{\mathrm{shell}}}$ = shell vertical bending inertia
M_r = root moment per vertical tail root chord
R_{fuse} = fuselage radius
S_{bulk} = bulkhead surface area
S_{nose} = nose surface area
V_{cone} = cone skin volume
V_{db} = web volume
W_{insul} = insulation material weight
W''_{insul} = weight/area density of insulation material
W_{shell} = shell weight
W_{skin} = skin weight
W_{web} = web weight
ΔP_{over} = cabin overpressure
ΔR_{fuse} = fuselage extension height
λ_{cone} = tailcone radius taper ratio
ρ_{skin} = skin density
σ_{skin} = max allowable skin stress
τ_{cone} = shear stress in tail cone
θ_{db} = double bubble fuselage joining angle
$c_{root_{\mathrm{vt}}}$ = vertical tail root chord
f_{fadd} = fractional added weight of local reinforcements
f_{frame} = fractional frame weight
f_{string} = fractional stringer weight
h_{db} = web half-height
h_{fuse} = fuselage height
l_{cone} = cone length
l_{shell} = shell length
t_{db} = web thickness
t_{shell} = shell thickness

t_{skin} = skin thickness
w_{db} = DB added half-width
w_{fuse} = fuselage half-width

2. Additional Constraints

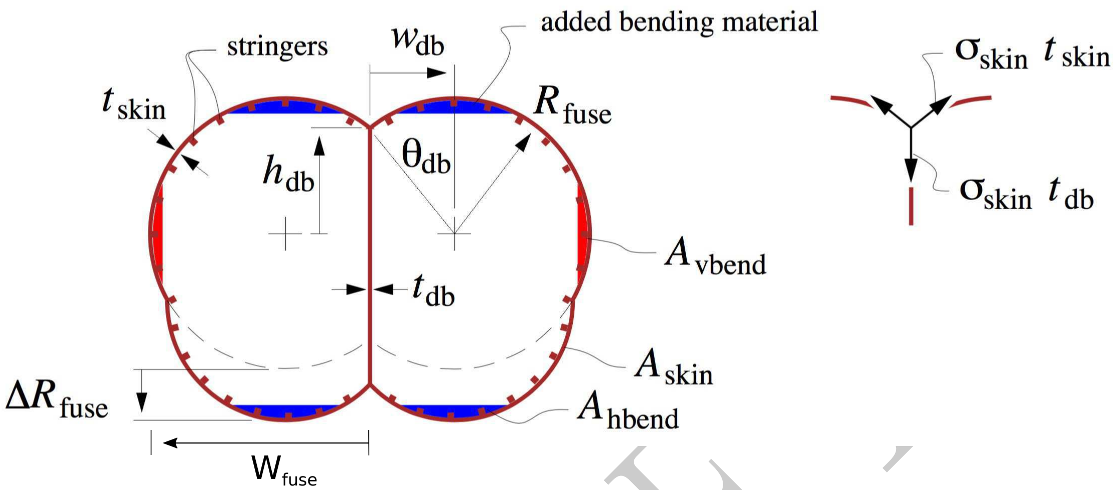


Fig. 9 Internal double bubble fuselage dimensions[1].

Figure 9 presents a cross sectional view of the double bubble fuselage. The added half-floor width due to the double bubble structure, w_{db}, is approximated with a first order Taylor expansion of the sine function.

$$\theta_{\text{db}} = \frac{w_{\text{db}}}{R_{\text{fuse}}} \tag{9}$$

A central tension web is added to account for the pressure forces in the fuselage center section. The web thickness depends on the internal pressure and the added floor half-width.

$$t_{\text{db}} = 2\frac{\Delta P_{over} w_{\text{db}}}{\sigma_{\text{skin}}} \tag{10}$$

The half-height of the web is lower bounded with a second order Taylor expansion of the cosine function.

$$-0.5 R_{\text{fuse}} {\theta_{\text{db}}}^2 + R_{\text{fuse}} \leq h_{\text{db}} \tag{11}$$

The half-width of the fuselage is incremented by the half-width of the central fuselage section.

$$R_{\text{fuse}} + w_{\text{db}} \geq w_{\text{fuse}} \tag{12}$$

A fuselage extension height, ΔR_{fuse}, augments the fuselage height and is constrained with a signomial equality constraint.

$$h_{\text{fuse}} = 0.5\Delta R_{\text{fuse}} + R_{\text{fuse}} \tag{13}$$

ΔR_{fuse} contributes to the shear web cross-sectional area and total material volume.

$$A_{\mathrm{db}} \geq 2h_{\mathrm{db}}t_{\mathrm{db}} + \Delta R_{\mathrm{fuse}}t_{\mathrm{db}} \tag{14}$$
$$V_{\mathrm{db}} = A_{\mathrm{db}}l_{\mathrm{shell}} \tag{15}$$

Web weight W_{db} is included in the total shell weight.

$$W_{\mathrm{db}} = V_{\mathrm{db}}\rho_{\mathrm{skin}}g \tag{16}$$
$$W_{\mathrm{shell}} \geq W_{\mathrm{db}} + W_{\mathrm{skin}} + W_{\mathrm{skin}}f_{\mathrm{fadd}} + W_{\mathrm{skin}}f_{\mathrm{frame}} + W_{\mathrm{skin}}f_{\mathrm{string}} \tag{17}$$

The skin cross sectional area, skin and bulkhead surface areas, and the tail cone volume are modified due to changing external geometry.

$$A_{\mathrm{skin}} \geq 2\Delta R_{\mathrm{fuse}}t_{\mathrm{skin}} + 4R_{\mathrm{fuse}}\theta_{\mathrm{db}}t_{\mathrm{skin}} + 2\pi R_{\mathrm{fuse}}t_{\mathrm{skin}} \tag{18}$$
$$S_{\mathrm{nose}} \geq 4{R_{\mathrm{fuse}}}^2\theta_{\mathrm{db}} + 2\pi {R_{\mathrm{fuse}}}^2 \tag{19}$$
$$S_{\mathrm{bulk}} \geq 4{R_{\mathrm{fuse}}}^2\theta_{\mathrm{db}} + 2\pi {R_{\mathrm{fuse}}}^2 \tag{20}$$
$$V_{\mathrm{cone}} \geq \frac{M_r c_{root_{\mathrm{vt}}}}{(1+\lambda_{\mathrm{cone}})\tau_{\mathrm{cone}}} \frac{\pi + 2\theta_{\mathrm{db}}}{\pi + 4\theta_{\mathrm{db}}} \frac{l_{\mathrm{cone}}}{R_{\mathrm{fuse}}} \tag{21}$$

The cross-sectional area of the fuselage, used for the calculation of cabin volume, is lower bounded as follows.

$$A_{\mathrm{fuse}} \geq -{R_{\mathrm{fuse}}}^2{\theta_{\mathrm{db}}}^3 + 2R_{\mathrm{fuse}}\Delta R_{\mathrm{fuse}} + \pi {R_{\mathrm{fuse}}}^2 + 4{R_{\mathrm{fuse}}}^2\theta_{\mathrm{db}} \tag{22}$$

The insulation weight constraint is incremented due to the increased surface area of the fuselage.

$$W_{\mathrm{insul}} \geq W''_{\mathrm{insul}}(1.1\pi + 2\theta_{\mathrm{db}})R_{\mathrm{fuse}}l_{\mathrm{shell}} + 0.55(S_{\mathrm{nose}} + S_{\mathrm{bulk}}) \tag{23}$$

The bending model, shown in Figure 10 and defined in [5], is modified due to the double bubble geometry and shear web, which provides additional bending reinforcement. These differences are captured in the bending area moments of inertia I_{hshell} and I_{vshell}.

$$\begin{aligned} I_{\mathrm{hshell}} \leq & \Bigg[(\pi + 4\theta_{\mathrm{db}})R_{\mathrm{fuse}}^2 + 8\left(1 - \frac{\theta_{\mathrm{db}}^2}{2}\right)(\frac{\Delta R_{\mathrm{fuse}}}{2})R_{\mathrm{fuse}} + \\ & (2\pi + 4\theta_{\mathrm{db}})(\frac{\Delta R_{\mathrm{fuse}}}{2})^2\Bigg]R_{\mathrm{fuse}}t_{\mathrm{shell}} + \frac{2}{3}\left[h_{\mathrm{db}} + \frac{\Delta R_{\mathrm{fuse}}}{2}\right]^3 t_{\mathrm{db}} \end{aligned} \tag{24}$$
$$I_{\mathrm{vshell}} \leq \left[\pi R_{\mathrm{fuse}}^2 + 8w_{\mathrm{db}}R_{\mathrm{fuse}} + (2\pi + 4\theta_{\mathrm{db}})w_{\mathrm{db}}^2\right]R_{\mathrm{fuse}}t_{\mathrm{shell}} \tag{25}$$

With the aforementioned modifications to the constraints from [5], the SP aircraft model can optimize both conventional tube and double bubble fuselages, with the fuselage joint angle θ_{db} adjusting the geometry.

D. Boundary Layer Ingestion

A boundary layer ingestion (BLI) model is required to model the D8.2. The D8.2 engine configuration is illustrated in Figure 11. As noted by Hall et al.[16], BLI on the D8.2 results in a reduction in required propulsor mechanical power of 9 percent. Three percent of the power savings comes from reduced jet dissipation while the remainder comes from a roughly three percent increase in propulsive efficiency and decreased airframe dissipation.

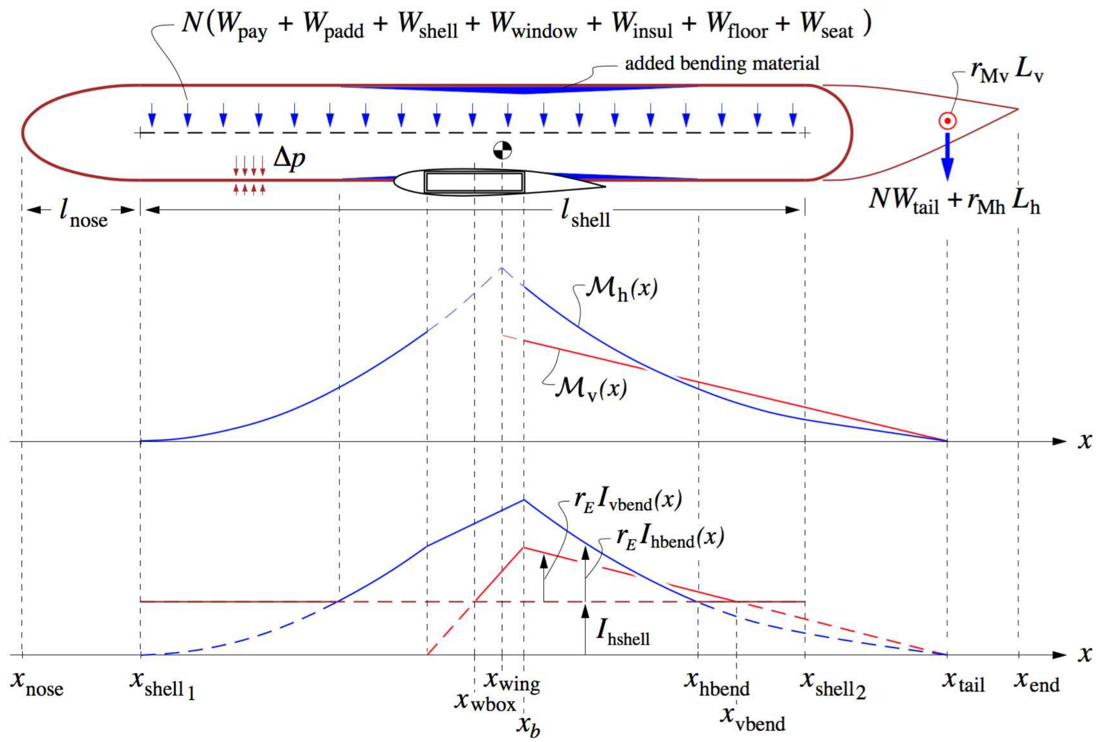


Fig. 10 TASOPT fuselage bending models[1]. The top graph depicts the bending load distribution and the bottom graph illustrates the shear load distribution. Horizontal bending loads are shown in blue, and the vertical bending loads are shown in red.

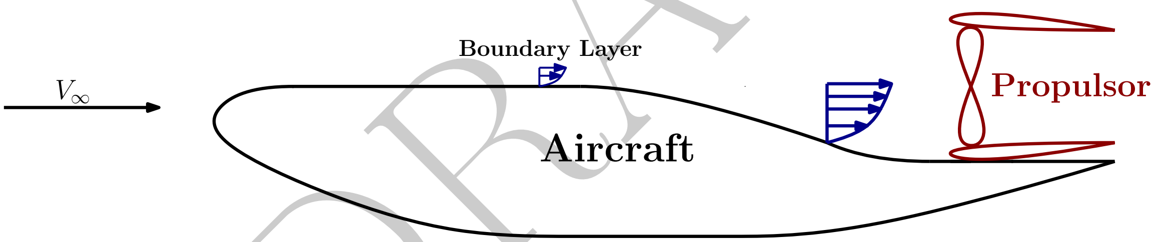


Fig. 11 Cartoon illustrating boundary layer growth on a BLI equipped aircraft similar to the D8.2.

1. BLI Terminology

D = drag
F = engine thrust
f_{BLI} = boundary layer ingestion fraction
$\text{f}_{\text{BLI}_\text{P}}$ = BLI induced engine inlet stagnation pressure loss factor
$\text{f}_{\text{BLI}_\text{V}}$ = BLI induced engine inlet velocity loss factor
f_{wake} = wake dissipation fraction
M_{min} = minimum cruise mach number
P = pressure
Φ = dissipation rate
ρ = density
u = engine working fluid velocity
$(\cdot)_{...0}$ = free stream quantity
$(\cdot)_{...6}$ = core exhaust quantity
$(\cdot)_{...8}$ = fan exhaust quantity
$(\cdot)_{...\text{atm}}$ = ambient atmospheric quantity

$(\cdot)_{\ldots t}$ = stagnation quantity

2. Fuselage Dissipation Model

The reduction in jet dissipation is modeled with a drag reduction factor, δ. Following Hall's[16] analysis, it is assumed the propulsor ingests 40 percent of the fuselage boundary layer ($f_{\text{BLI}} = 0.4$). It is further assumed one third of total dissipation (Φ) is surface dissipation (Φ_{surf}). The wake dissipation fraction, f_{wake}, is defined by Equation 26 and assumed equal to 0.08. After noting $\Phi = DV_\infty$, this analysis yields Equations 27 and 28.

$$f_{\text{wake}} = \frac{\Phi_{\text{wake}}}{\Phi_{\text{wake}} + \Phi_{\text{surf}}} \tag{26}$$

$$\delta = f_{\text{BLI}} * 0.33 * 0.08 \tag{27}$$

$$D_{\text{total}} = \delta(D_{\text{induced}} + D_{\text{airframe}}) \tag{28}$$

3. Engine Boundary Layer Ingestion

BLI engines ingest air with lower average velocity, and in turn lower stagnation pressure, than free stream air. Three constraints from [6] were modified to account for BLI. Engine inlet stagnation pressure was reduced by the factor $\mathrm{f_{BLI_P}}$. Note $\mathrm{f_{BLI_P}}$ represents the average drop in stagnation pressure across the entire inlet. Following [6], Z_0 replaces the non-GP compatible expression $1 + \frac{\gamma-1}{2}(M_0)^2$ in stagnation relations.

$$P_{t_0} = \mathrm{f_{BLI_P} P_{atm} Z_0}^{3.5} \tag{29}$$

Thrust is equal to the working fluid's rate of momentum change. The factor $\mathrm{f_{BLI_V}}$ was introduced to fan and core thrust constraints to account for the decrease in average free stream velocity. Again, $\mathrm{f_{BLI_V}}$ is the average velocity drop across the entire fan.

$$\frac{F_8}{\alpha \dot{m}_{\text{core}}} + \mathrm{f_{BLI_V} u_0} \leq u_8 \tag{30}$$

$$\frac{F_6}{\bar{f}_o \dot{m}_{\text{core}}} + \mathrm{f_{BLI_V} u_0} \leq u_6 \tag{31}$$

Determining $\mathrm{f_{BLI_P}}$ and $\mathrm{f_{BLI_V}}$ can be difficult. As of now, there are no GP or SP compatible boundary layer models so either $\mathrm{f_{BLI_P}}$ or $\mathrm{f_{BLI_V}}$ must be estimated. Using the experimental results presented by Hall et al.[17] $\mathrm{f_{BLI_V}}$ was estimated to be 0.0727. $\mathrm{f_{BLI_P}}$ was then determined using Equation 32.

$$\mathrm{f_{BLI_P}} = \frac{\mathrm{P_{atm}} + \rho_{\text{atm}}(\mathrm{f_{BLI_V} M_{min} a})^2}{\mathrm{P_{atm}} + \rho_{\text{atm}}(\mathrm{M_{min} a})^2} \tag{32}$$

Finally, it is important to note BLI fan distortion effects will decrease fan efficiency to approximately 90%[18].

E. Horizontal Tail Structural Model Modifications

An update to the structural model in [5] was required to accurately model the bending and shear loads on horizontal pi-tails. This section derives and presents a new set of constraints, which are compatible with both conventional tail and pi-tail architectures.

1. Assumptions

1. The lift per unit span is proportional to local chord.

2. The horizontal tail has a constant taper ratio.

3. The horizontal and vertical tail joint is a fuselage width away from the centerline of the aircraft.

4. The horizontal and vertical tail interface is a pin joint. Therefore, the joint does not exert a moment on the horizontal tail.

5. The shear and moment distributions on the horizontal tail are linearized.

The pin-joint assumption ensures the vertical tail structural constraints do not need to be modified for the pi-tail configuration.

2. Sample Free Body Diagram and Load Distributions

With the aforementioned assumptions, the free body diagram diagram of the pi-tail is shown at the top of Figure 12.

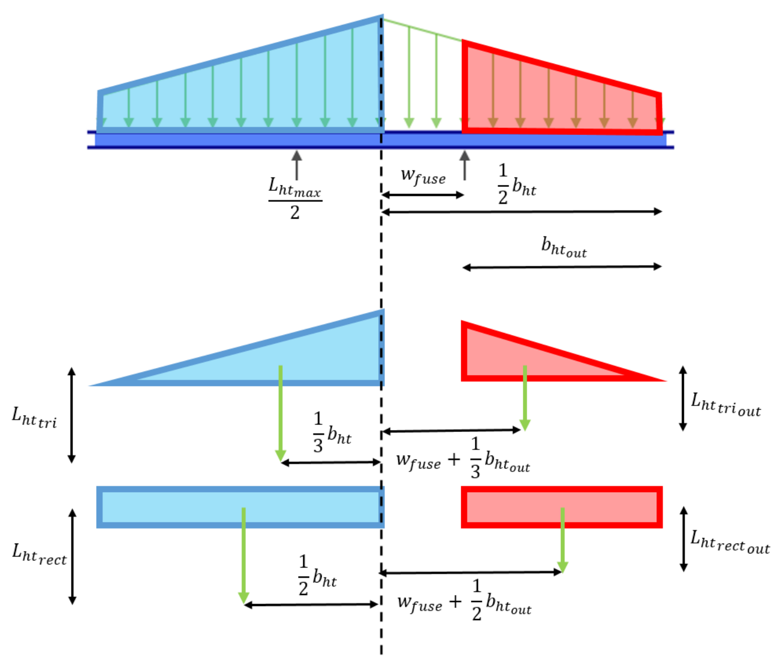


Fig. 12 Free body diagram of the forces on the horizontal tail. The distributed lift force, which is assumed to be proportional to local chord, is partitioned into triangular and rectangular components.

Shear and moment diagrams are presented in Figures 13 and 14 respectively. The diagrams include both the distributed lift loads (green arrows in Figure 12) and the point loads of imposed on the pin joints by the vertical tails.

3. Horizontal Tail Terminology

I_{cap} = non-dimensional spar cap area moment of inertia
L_{ht} = horizontal tail downforce
$L_{\mathrm{ht_{max}}}$ = maximum horizontal tail downforce
$L_{\mathrm{ht_{rect}}}$ = rectangular horizontal tail load
$L_{\mathrm{ht_{rect_{out}}}}$ = rectangular horizontal tail load outboard

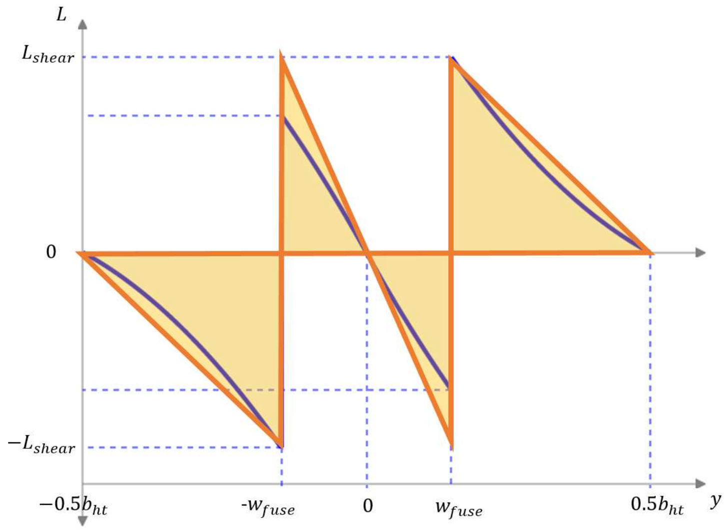


Fig. 13 Shear diagram of the pi-tail. The blue line shows the actual loading, while the yellow line with infill shows the assumed load distribution.

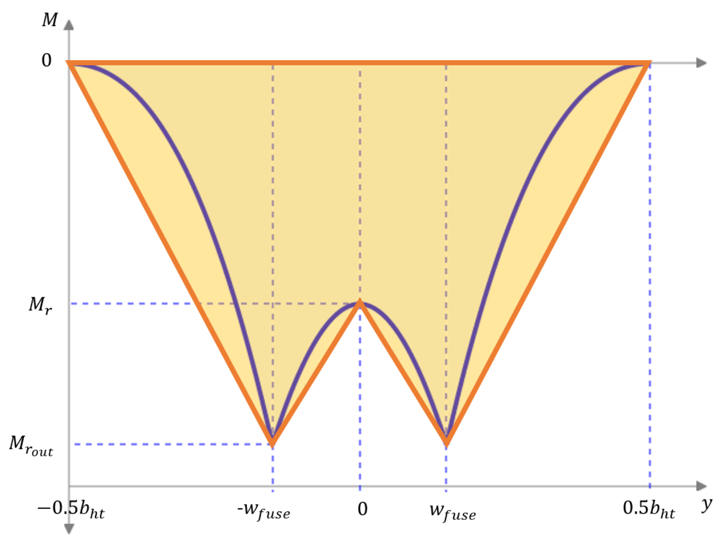


Fig. 14 Moment diagram of the pi-tail. The blue line shows the actual loading, and the yellow line shows the assumed load distribution

$L_{\mathrm{ht_{tri}}}$ = triangular horizontal tail load
$L_{\mathrm{ht_{tri_{out}}}}$ = triangular horizontal tail load outboard
L_{shear} = maximum shear load at pin-joint
M_{r} = moment per chord at horizontal tail root
$M_{\mathrm{r_{out}}}$ = moment per chord at pin-joint
N_{lift} = horizontal tail loading multiplier
S_{ht} = horizontal tail area
W_{cap} = weight of spar caps
W_{struct} = horizontal tail wingbox weight
W_{web} = weight of shear web
λ_{ht} = horizontal tail taper ratio
ν = dummy variable = $(t^2 + t + 1)/(t + 1)^2$
$\pi_{\mathrm{M-fac}}$ = pi-tail bending structural factor

ρ_{cap} = density of spar cap material
ρ_{web} = density of shear web material
$\sigma_{max,shear}$ = allowable shear stress
$\sigma_{\max}$ = allowable tensile stress
τ_{ht} = horizontal tail thickness/chord ratio
b_{ht} = horizontal tail span
$b_{\text{ht}_{\text{out}}}$ = horizontal tail outboard half-span
c_{attach} = horizontal tail chord at the pin-joint
$c_{\text{root}_{\text{ht}}}$ = horizontal tail root chord
$c_{\text{tip}_{\text{ht}}}$ = horizontal tail tip chord
g = gravitational acceleration
q_{ht} = substituted variable = 1 + taper
r_h = fractional wing thickness at spar web
t_{cap} = non-dim. spar cap thickness
t_{web} = non-dim. shear web thickness
w = wingbox width-to-chord ratio
w_{fuse} = fuselage half-width

4. Load Derivation

$L_{\text{ht}_{\text{rect}}}$ is defined to be half the lift generated by the 'rectangular' section of the wing (the blue rectangle in Figure 12).

$$L_{\text{ht}_{\text{rect}}} \geq \frac{L_{\text{ht}_{\max}} c_{\text{tip}_{\text{ht}}} b_{\text{ht}}}{2S_{\text{ht}}} \tag{33}$$

Similarly, $L_{\text{ht}_{\text{tri}}}$ is defined to be half the lift generated by the 'triangular' section of the wing (the blue triangle in Figure 12).

$$L_{\text{ht}_{\text{tri}}} \geq \frac{L_{\text{ht}_{\max}} (1-\lambda_{\text{ht}}) c_{\text{root}_{\text{ht}}} b_{\text{ht}}}{4S_{\text{ht}}} \tag{34}$$

After defining the horizontal tail half-span outboard of the pin joint ($b_{\text{ht}_{\text{out}}}$), the outboard components of the lift loads can be computed with respect to $L_{\text{ht}_{\text{rect}}}$ and $L_{\text{ht}_{\text{tri}}}$. The outboard loads are shown in red in Figure 12.

$$b_{\text{ht}_{\text{out}}} \geq 0.5 b_{\text{ht}} - w_{\text{fuse}} \tag{35}$$

$$L_{\text{ht}_{\text{tri}_{\text{out}}}} \geq L_{\text{ht}_{\text{tri}}} \frac{b_{\text{ht}_{\text{out}}}}{(0.5 b_{\text{ht}})^2} \tag{36}$$

$$L_{\text{ht}_{\text{rect}_{\text{out}}}} \geq L_{\text{ht}_{\text{rect}}} \frac{b_{\text{ht}_{\text{out}}}}{0.5 b_{\text{ht}}} \tag{37}$$

The horizontal-vertical tail pin joint is assumed to be exactly at w_{fuse}. This is a conservative estimate. In most pi-tail configurations the vertical tails are canted outwards. The local chord at the pin joint is constrained with the following monomial equality.

$$c_{\text{attach}} = \frac{b_{\text{ht}} \lambda_{\text{ht}} c_{\text{root}_{\text{ht}}}}{2 w_{\text{fuse}}} \tag{38}$$

The maximum moment at the joint is determined by summing the bending moment contributions from loads outboard of the joint.

$$M_{r_{\text{out}}} c_{\text{attach}} \geq L_{\text{ht}_{\text{rect}_{\text{out}}}} \frac{1}{2} b_{\text{ht}_{\text{out}}} + L_{\text{ht}_{\text{tri}_{\text{out}}}} \frac{1}{3} b_{\text{ht}_{\text{out}}}, \tag{39}$$

The maximum shear at the joint is the sum of the outboard shear loads. The maximum root moment is the sum of the bending loads from lift and the pin-joint load.

$$L_{\mathrm{shear}} \geq L_{\mathrm{ht}_{\mathrm{rect}_{\mathrm{out}}}} + L_{\mathrm{ht}_{\mathrm{triout}}} \tag{40}$$

$$M_{\mathrm{r}} c_{\mathrm{root}_{\mathrm{ht}}} \geq L_{\mathrm{ht}_{\mathrm{rect}}} \frac{1}{4} b_{\mathrm{ht}} + L_{\mathrm{ht}_{\mathrm{tri}}} \frac{1}{6} b_{\mathrm{ht}} - \frac{1}{2} L_{\mathrm{ht}_{\mathrm{max}}} w_{\mathrm{fuse}} \tag{41}$$

Finally, the wingtip moment is set equal to zero with a signomial equality constraint.

$$\frac{b_{\mathrm{ht}}}{4} L_{\mathrm{ht}_{\mathrm{rect}}} + \frac{b_{\mathrm{ht}}}{3} L_{\mathrm{ht}_{\mathrm{tri}}} = b_{\mathrm{ht}_{\mathrm{out}}} \frac{L_{\mathrm{ht}_{\mathrm{max}}}}{2} \tag{42}$$

5. *Structural Sizing*

Equations from [3] for wing structural sizing were adapted using a linearization of the moment and shear load distributions from Appendix E2. The constraints can be applied to both conventional and pi-tails.

$$0.92 w \tau_{\mathrm{ht}} t_{\mathrm{cap}}^2 + I_{\mathrm{cap}} \leq \frac{0.92^2}{2} w \tau_{\mathrm{ht}}^2 t_{\mathrm{cap}} \tag{43}$$

$$8 \geq N_{\mathrm{lift}} M_{\mathrm{r}_{\mathrm{out}}} (AR_{\mathrm{ht}}) q_{\mathrm{ht}}^2 \frac{\tau_{\mathrm{ht}}}{S_{\mathrm{ht}} I_{\mathrm{cap}} \sigma_{\mathrm{max}}} \tag{44}$$

$$12 \geq \frac{2 L_{\mathrm{shear}} N_{\mathrm{lift}} q^2}{\tau_{\mathrm{ht}} S t_{\mathrm{web}} \sigma_{\mathrm{max-shear}}} \tag{45}$$

The changes to the model in [3] are:

- In the shear constraint replacing $L_{\mathrm{ht}_{\mathrm{max}}}$ with $2L_{\mathrm{shear}}$. This is done because the shear loads for the pi-tail are different than the maximum lift loads for the conventional tail.
- Replacing M_{r} with $M_{\mathrm{r}_{\mathrm{out}}}$, the moment per unit chord at the pin joint. For a pi-tail, maximum bending loads occur at the pin joint.

The linearization of the shear and bending load distributions simplifies the derivation of the structural web and cap weights. Shear web sizing relies on the assumption that the maximum shear (L_{shear}) occurs at the pin-joint and the weight of the shear web of the pi-tail under L_{shear} is equal to the shear web weight of a conventional tail subjected to the the same maximum shear load at its root. This is a conservative approximation, the load distribution implied by this assumption (shown in yellow in Figure 13) has a larger internal area than the actual load distribution. Intuitively, the L_{shear} for a pi-tail is strictly smaller than the L_{shear} a conventional tail of the same size and loading. The pi-tail more efficient in shear.

The cap weight of the pi-tail is determined by scaling the cap weight of a conventional tail with the same geometry as the pi-tail and a root moment of $M_{\mathrm{r}_{\mathrm{out}}} c_{\mathrm{attach}}$. The scaling factor, $\pi_{\mathrm{M-fac}}$, is the ratio of the total shaded bending moment area in Figure 14 to the sum of the outboard shaded areas multiplied by the ratio of the outboard half-span to the total half-span.

$$\pi_{\mathrm{M-fac}} \geq \left[\frac{\frac{1}{2}(M_{\mathrm{r}_{\mathrm{out}}} c_{\mathrm{attach}} + M_{\mathrm{r}} c_{\mathrm{root}_{\mathrm{ht}}}) w_{\mathrm{fuse}}}{\frac{1}{2} M_{\mathrm{r}_{\mathrm{out}}} c_{\mathrm{attach}} b_{\mathrm{ht}_{\mathrm{out}}}} + 1.0 \right] \frac{b_{\mathrm{ht}_{\mathrm{out}}}}{0.5 b_{\mathrm{ht}}}, \tag{46}$$

Given the calculated loads and structural factors, the bending material and shear web weight can be calculated.

$$W_{\mathrm{cap}} \geq \frac{\pi_{\mathrm{M-fac}} 8 \rho_{\mathrm{cap}} g w t_{\mathrm{cap}} S_{\mathrm{ht}}^1 .5 \nu}{3 AR_{\mathrm{ht}}^{0.5}} \tag{47}$$

$$W_{\mathrm{web}} \geq \frac{8 \rho_{\mathrm{web}} g r_h \tau_{\mathrm{ht}} t_{\mathrm{web}} S_{\mathrm{ht}}^{1.5} \nu}{3 AR_{\mathrm{ht}}^{0.5}} \tag{48}$$

$$W_{\mathrm{struct}} \geq W_{\mathrm{web}} + W_{\mathrm{cap}} \tag{49}$$

The value for t_{cap} is notional in the derivation above. Rather than being the spar cap thickness of a pi-tail, it is the spar cap thickness required for a conventional tail of the same geometry and a root moment ($M_{r_{\text{out}}} c_{\text{attach}}$) as a pi-tail. With a similar reasoning as for the shear loads, $\pi_{\text{M-fac}} t_{\text{cap}}$ for a pi-tail is strictly smaller than the t_{cap} for a conventional tail of the same geometry and loading, making the pi-tail more efficient in bending than a traditional tail.

F. Mission Profile

The mission profile includes weight, drag, and altitude build up constraints as well as a series of aircraft performance constraints. The mission profile can be discretized into an arbitrary number of climb and cruise segments. The profile allows for the possibility of a cruise climb. The descent portion of the flight was neglected due to the small percent of mission time and fuel burn it encompasses. Neglecting descent results in a slight over-estimation of total mission fuel burn.

1. *mission profile Terminology*

a = speed of sound
D = total aircraft drag
$D_{\text{components}}$ = drag on aircraft subsystems
D_{induced} = induced drag
Δh = altitude change
F = engine thrust
$f_{\text{fuel}_{\text{res}}}$ = reserve fuel fraction
$h_{\text{cruise,min}}$ = minimum cruise altitude
L = sum of wing and fuselage lift
L_{ht} = horizontal tail down force
$\frac{L}{D}$ = Aircraft lift to drag ratio
M = mach number
M_{min} = minimum cruise mach number
N_{eng} = aircraft's number of engines
θ = climb angle
h = altitude
P_{excess} = excess power
R = downrange distance covered
R_{req} = total required range
RC = rate of climb
t = flight segment duration
$t_{\text{climb,max}}$ = max allowed time to climb
TSFC= thrust specific fuel consumption
V = aircraft speed
W = takeoff weight
W_{buoy} = buoyancy force
W_{avg} = average flight segment aircraft weight
W_{dry} = aircraft dry weight
W_{end} = aircraft flight segment end weight
W_{engine} = engine weight
W_{fuel} = flight segment fuel weight burned
W_{fuse} = fuselage weight
W_{ht} = horizontal tail weight
W_{lg} = landing gear weight
W_{misc} = miscellaneous system weight
W_{payload} = payload weight
$W_{f_{\text{primary}}}$ = total fuel weight less reserves
$W_{\text{fuel}_{\text{total}}}$ = total fuel weight
W_{start} = aircraft flight segment start weight
W_{vt} = vertical tail weight

W_{wing} = wing weight
$(\cdot)_{0..i..N}$ = flight segment i quantity

2. *Weight and Drag Build Ups*

Downward optimization pressure on weight and drag allows basic posynomial weight and drag build ups to be used.

$$D_i \geq \sum D_{\text{components}_\text{i}} + D_{\text{induced}_\text{i}} \tag{50}$$

$$\tag{51}$$

W_{avg_i} is the geometric mean of a segments start and end weight. Average weight is used instead of either the segment start or end weight. This increases model accuracy and stability.

$$W_{\text{dry}} \geq W_{\text{wing}} + W_{\text{fuse}} + W_{\text{vt}} + W_{\text{ht}} + W_{\text{lg}} + W_{\text{eng}} + W_{\text{misc}} \tag{52}$$

$$\sum_{i=1}^{N} W_{\text{fuel}_\text{i}} \leq W_{f_{\text{primary}}} \tag{53}$$

$$W \geq W_{\text{dry}} + W_{\text{payload}} + f_{\text{fuel}_{\text{res}}} W_{f_{\text{primary}}} \tag{54}$$

$$W_{\text{start}_\text{i}} \geq W_{\text{end}_\text{i}} + \sum_{n=1}^{i} W_{\text{fuel}_\text{n}} \tag{55}$$

$$W_{\text{start}_0} = W \tag{56}$$

$$W_{\text{end}_\text{N}} \geq W_{\text{dry}} + W_{\text{payload}} + f_{\text{fuel}_{\text{res}}} W_{f_{\text{primary}}} \tag{57}$$

$$W_{\text{start}_{\text{i+1}}} = W_{end_i} \tag{58}$$

$$W_{\text{avg}_\text{i}} \geq \sqrt{W_{\text{start}_\text{i}} W_{\text{end}_\text{i}}} + W_{\text{buoy}_\text{i}} \tag{59}$$

$$\tag{60}$$

3. *General Performance Constraints*

The sum of segment ranges is constrained to be greater than or equal to the required range.

$$\sum_{i=1}^{N} R_i \geq R_{\text{req}} \tag{61}$$

$$\tag{62}$$

Segment fuel burn is a function of TSFC, thrust, and segment flight time.

$$W_{\text{fuel}_i} = N_{\text{eng}} \text{TSFC}_i t_i F_i \tag{63}$$

Altitude change during each segment is a function of climb rate and total segment time. Equation 65 uses a small angle approximation to compute the downrange distance covered during each segment.

$$\Delta h_i = t_i \text{RC}_i \tag{64}$$

$$t_i V_i = \text{Range}_i \tag{65}$$

Standard lift to drag and Mach number definitions are used.

$$M_i = \frac{V_i}{a_i} \tag{66}$$

$$\left(\frac{L}{D}\right)_i = \frac{W_{avg_i}}{D_i} \tag{67}$$

4. Climb Performance Constraints

Climb rates are computed with an excess power formulation[19]. During the first climb segment, the climb rate is constrained to be greater than 2,500 ft/min. For all remaining climb segments, the climb rate is constrained to be greater than 500 ft/min. The climb angle, θ, is set using a small angle approximation.

$$P_{\text{excess}} + V_i D_i \leq V_i N_{\text{eng}} F_i \tag{68}$$

$$\text{RC}_i = \frac{P_{\text{excess}}}{W_{\text{avg}_i}} \tag{69}$$

$$\theta_i V_i = \text{RC}_i \tag{70}$$

There can be either an upward or downward pressure on h. Thus, a signomial equality constraint must be used to constrain altitude.

$$h_i = h_{i-1} + \Delta h_i \tag{71}$$

In the above formulation, h_i is equivalent to segment end altitude.

$$h_0 = \Delta h_0 \tag{72}$$

Climb segments are constrained to have equal altitude changes and the final climb segment altitude is constrained to be greater than a user specified minimum cruise altitude. If no minimum cruise altitude is specified, $h_{\text{cruise,min}}$ is optimized.

$$\Delta h_{i+1} = \Delta h_i \tag{73}$$

$$h_{\text{N}_{\text{climb}}} \geq h_{\text{cruise,min}} \tag{74}$$

Time to climb is constrained to be less than a user specified maximum value. If no maximum value is specified, $t_{\text{climb}_{\text{max}}}$ is optimized.

$$\sum_{0}^{N_{\text{climb}}} t_i <= t_{\text{climb,max}} \tag{75}$$

Finally, the climb gradient at top of climb is constrained to be greater than 0.015 radians.

$$\theta_{\text{N}_{\text{climb}}} \geq 0.015 \tag{76}$$

5. Cruise Performance Constraints

Cruise range segments are constrained to be equal length. Cruise Mach number is constrained to be greater than a user specified minimum. If no minimum is specified, M_{min} is optimized.

$$R_{i+1} = R_i \tag{77}$$

$$M_i \geq M_{\text{min}} \tag{78}$$

The cruise climb angle is assumed to be small. The sum of wing and fuselage lift is set equal to weight plus horizontal tail down force. Thrust must overcome both drag and the portion of aircraft weight acting in the direction of thrust. These constraints are a conservative approximation of flight physics.

$$L_i \geq W_{\mathrm{avg_i}} + L_{\mathrm{ht_i}} \tag{79}$$

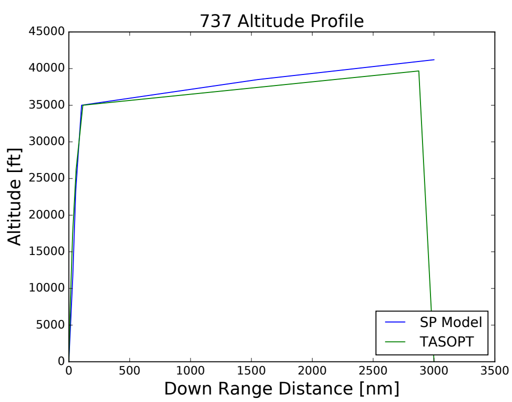
$$N_{\mathrm{eng}} F_i \geq D_i + W_{\mathrm{avg_i}} \theta_i \tag{80}$$

In cruise, there is a downward pressure on segment end altitude, removing the need for a signomial equality.

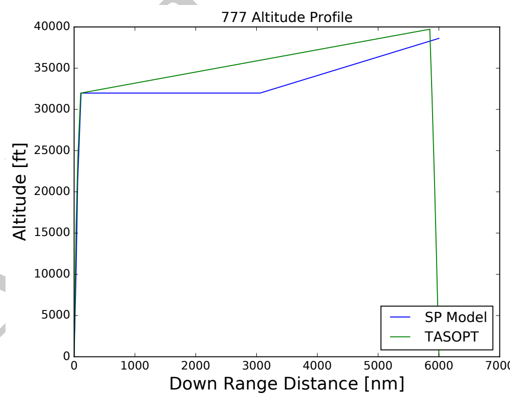
$$h_i \geq h_{i-1} + \Delta h_i \tag{81}$$

6. Example mission profiles

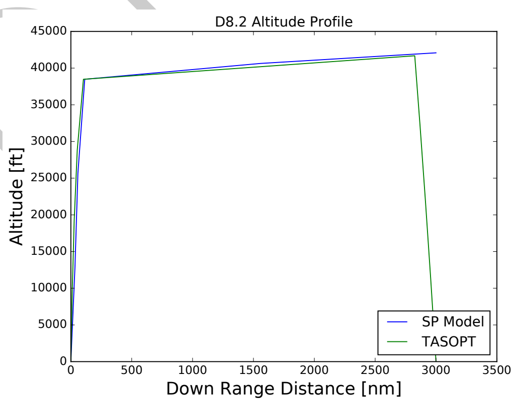
Figure 15 presents the 737, 777, and D8.2 mission profiles generated by the SP tool overlaid with TASOPT mission profiles.



(a) 737 flight profile.



(b) 777 mission profile. The SP 777 does not cruise climb until the second cruise segment.



(c) D8.2 flight profile.

Fig. 15 SP and TASOPT mission profiles for the 737, 777, and D8.2 models presented in Section III D..

G. Sensitivity Overview

A strength of convex and difference of convex optimization is that, together with the optimum solution, it provides sensitivities of the objective function to all model parameter values and constraints. Sensitivities are all local and computed about the optimum found in the last GP approximation of the SP. Equation 1 is the formula for parameter sensitivities while Equation 2 is the formula for constraint sensitivities[10]. GPkit computes sensitivities via Lagrange duality using the method developed by Hoburg [4]. If the sensitivity to a constant is 0.5 then decreasing that constant by one percent will decrease the objective by approximately one half a percent. If the sensitivity to a constant is -0.75, then a one percent increase in the constant will decrease the objective by approximately three quarters of a percent. If the sensitivity to a constraint is 0.9, then loosing that constraint by one percent will decrease the objective function by nine-tenths of a percent. Tightening the constraint by one percent would increase the objective function by nine-tenths of a percent. When using the relaxed constants SP solution heuristic in [5], sensitivities are computed relative to the original (non-relaxed) value of each parameter.

Acknowledgments

This work was partially funded by Aurora Flight Sciences.

References

[1] Drela, M., "N3 Aircraft Concept Designs and Trade Studies - Appendix. Technical Report NASA CR-2010-216794/VOL2," Tech. rep., NASA, 2010.

[2] Martins, J. R. and Lambe, A. B., "Multidisciplinary Design Optimization: A survey of Architectures," *AIAA Journal*.

[3] Hoburg, W. and Abbeel, P., "Geometric Programming for Aircraft Design Optimization," *AIAA Journal*.

[4] Hoburg, W., *Aircraft Design Optimization as a Geometric Program*, Ph.D. thesis, University of California Berkeley, 2013.

[5] Kirschen, P. G., York, M. A., Ozturk, B., and Hoburg, W. W., "Application of Signomial Programming to Aircraft Design," *AIAA Journal of Aircraft*.

[6] York, M., Hoburg, W., and Drela, M., "Turbofan Engine Sizing and Tradeoff Analysis via Signomial Programming," *AIAA Journal of Aircraft*.

[7] Drela, M., "Development of the D8 Transport Configuration," 29th AIAA Applied Aerodynamics Conference, The address of the publisher, 2011.

[8] Burnell, E. and Hoburg, W., "GPkit software for geometric programming," `https://github.com/hoburg/gpkit`, 2015. Version 0.4.0.

[9] ApS, "The MOSEK C optimizer API manual," , 2015. Version 7.1 (Revision 41).

[10] Boyd, S., Kim, S.-J., Vanderberghe, L., and Hassibi, A., "A tutorial on geometric programming," *Optim Eng*.

[11] open source, N., "OpenVSP," `http://www.openvsp.org/`, 2017. Version 3.11.0.

[12] Duffin, R., Peterson, E., and Zener, C., *Geometric programming: theory and application,*, Wiley New York, 1967.

[13] Boyd, S. and Vandenberghe, L., *Convex Optimization*, Cambridge University Press, The address, 7th ed., 2009.

[14] Lipp, T. and Boyd, S., "Variations and extension of the convex concave procedure," *Optim Eng*.

[15] Opgenoord, M. M. J., Cohen, B. S., and Hoburg, W. W., "Comparison of Algorithms for Including Equality Constraints in Signomial Programming," Tech. Rep. TR-17-1, Aerospace Computational Design Laboratory (ACDL), Massachusetts Institute of Technology (MIT), Cambridge, Massachusetts, 2017.

[16] Hall, D. K., Huang, A. C., Uranga, A., Greitzer, E. M., Drela, M., and Sato, S., "Boundary Layer Ingestion Propulsion Benefit for Transport Aircraft," *AIAA Journal of Propulsion and Power*.

[17] D. K. Hall, E. M. G. and Tan, C. S., "Analysis of Fan Stage Conceptual Design Attributes for Boundary Layer Ingestion," *Journal of Turbomachinery*.

[18] Plas, A., Sargeant, M., Madani, V., Crichton, D., andT. Hynes, E. G., and Hall, C., "Performance of a Boundary Layer Ingesting (BLI) Propulsion System," *AIAA Aerospace Sciences Meeting and Exhibit*.

[19] Anderson, J. D., *Aircraft Performance and Design*, WCB McGraw-Hill, Boston, Massachusetts, 1999.