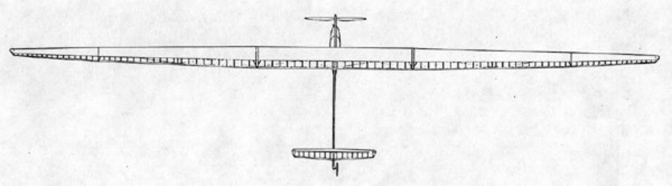
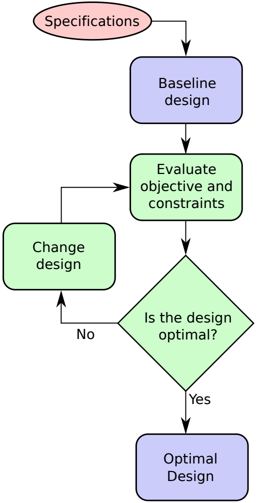


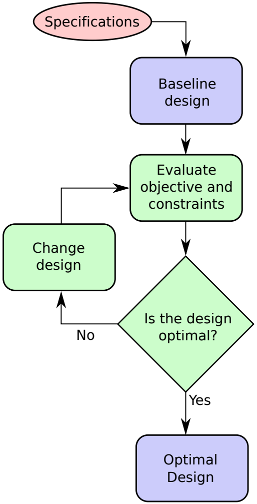
Exploiting convex structure in aircraft design optimization

Warren Hoburg
Massachusetts Institute of Technology
with Philippe Kirschen, Cody Karcher, and Ned Burnell

NASA Ames Applied Modeling and Simulation Seminar
June 5, 2015

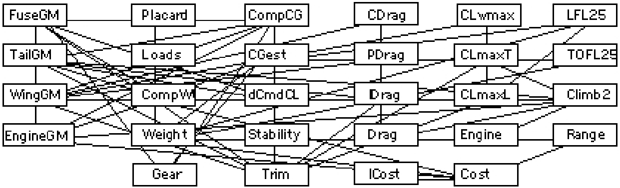


[Adapted from Alonso 2012]



[Adapted from Alonso 2012]

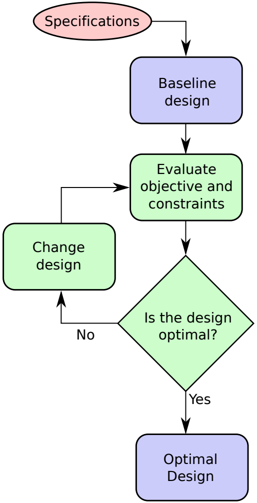
Challenge #1: Black box models

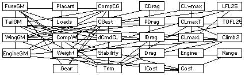


[Kroo 1994]

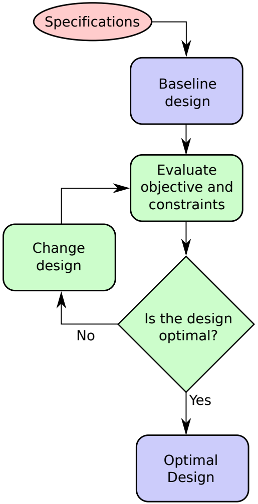
- Disciplinary separations
- Expensive function evaluations

Challenge #1: Black box models

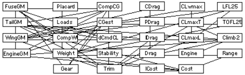




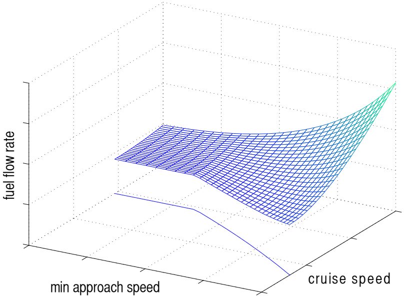
[Adapted from Alonso 2012]



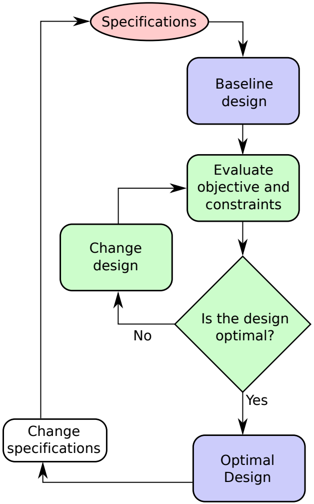
Challenge #1: Black box models



Challenge #2: Design tradeoffs

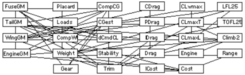


[Adapted from Alonso 2012]

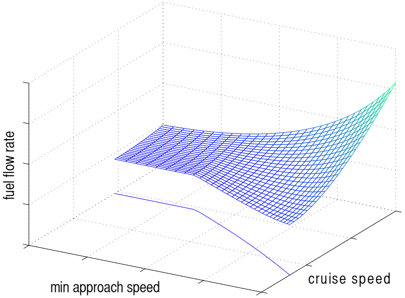


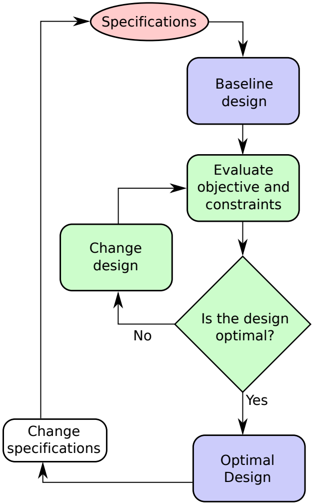
[Adapted from Alonso 2012]

Challenge #1: Black box models



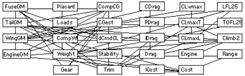
Challenge #2: Design tradeoffs





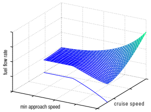
[Adapted from Alonso 2012]

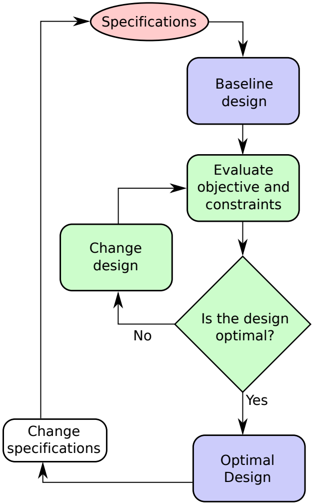
Challenge #1: Black box models



Challenge #2: Design tradeoffs

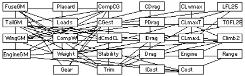
- Want Pareto frontier
  $\rightarrow$ must solve many times





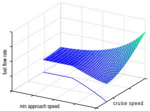
[Adapted from Alonso 2012]

Challenge #1: Black box models



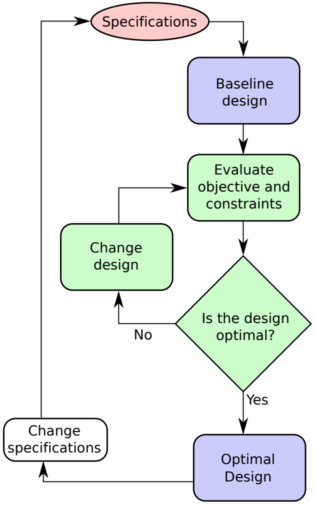
Challenge #2: Design tradeoffs

- Want Pareto frontier
  → must solve many times

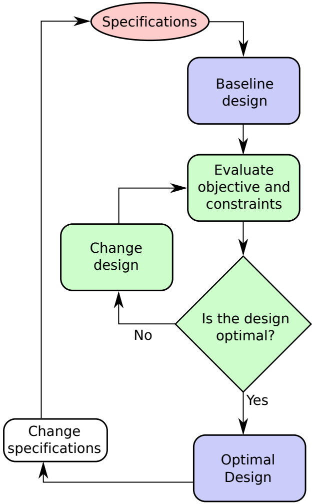


Challenge #3: Optimization issues

- Local vs. global optima
- Sensitivity to initial guess
- Sensitivity to solver params



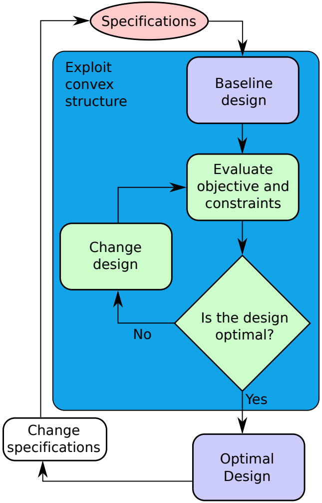
[Adapted from Alonso 2012]



[Adapted from Alonso 2012]

Insight

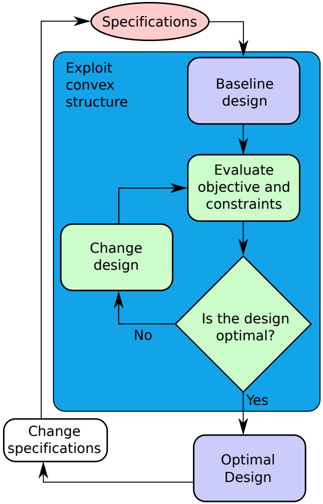
- Surprisingly, many relationships in engineering design have an underlying convex structure.



[Adapted from Alonso 2012]

Insight

- Surprisingly, many relationships in engineering design have an underlying convex structure.



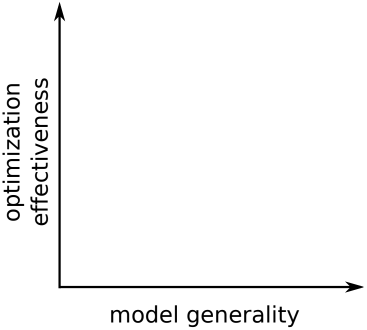
[Adapted from Alonso 2012]

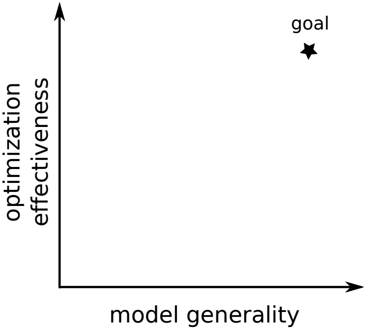
Insight

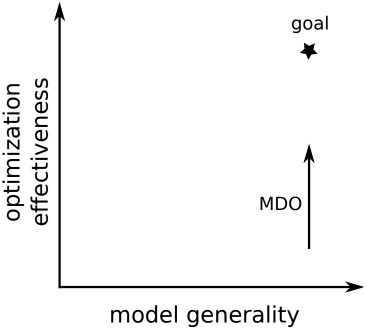
- Surprisingly, many relationships in engineering design have an underlying convex structure.

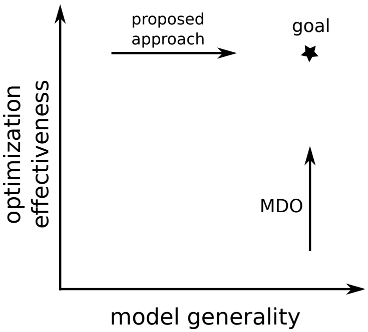
Benefits

- Globally optimal solutions
- Numerically stable algorithms
- No initial guesses; no solver parameter tuning
- Extremely fast solutions, even for large problems


optimization effectiveness
model generality


goal
optimization effectiveness
model generality


goal
MDO
optimization effectiveness
model generality


proposed
approach
goal
MDO
optimization
effectiveness
model generality

Today's talk

Geometric programming overview

Aircraft design modeling examples

Current and future directions

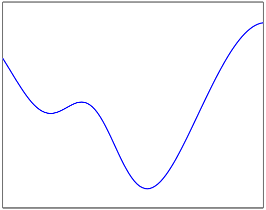
Today's talk

Geometric programming overview

Aircraft design modeling examples

Current and future directions

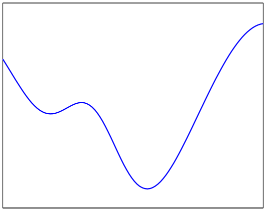
General nonlinear program



$$
\begin{aligned}
\text{minimize} \quad & f_0(\boldsymbol{x}) \\
\text{subject to} \quad & f_i(\boldsymbol{x}) \leq 0, \quad i = 1, ..., N_i \\
& h_j(\boldsymbol{x}) = 0, \;\; j = 1, ..., N_e
\end{aligned}
$$

- In general, extremely difficult to solve

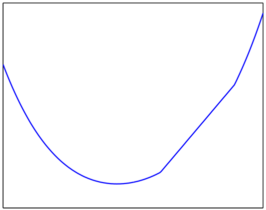
General nonlinear program



$$
\begin{aligned}
\text{minimize} \quad & f_0(\boldsymbol{x}) \\
\text{subject to} \quad & f_i(\boldsymbol{x}) \le 0, \quad i = 1, \dots, N_i \\
& h_j(\boldsymbol{x}) = 0, \quad j = 1, \dots, N_e
\end{aligned}
$$

- In general, extremely difficult to solve

Convex program



Same as nonlinear program, except

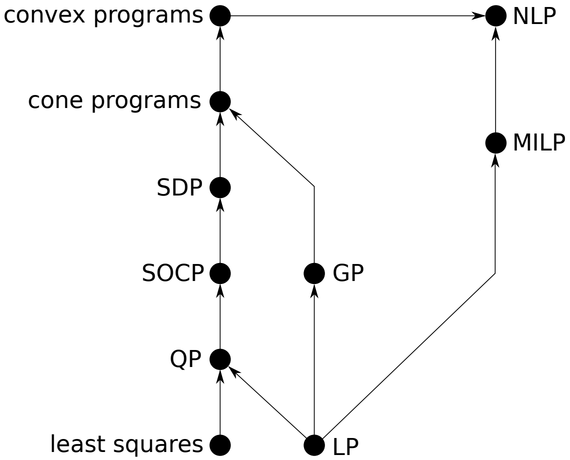
- $f_i(\boldsymbol{x})$ must be convex
- $h_j(\boldsymbol{x})$ must be affine

- Very efficient to solve

"... the great watershed in optimization isn't between linearity and nonlinearity, but convexity and nonconvexity."

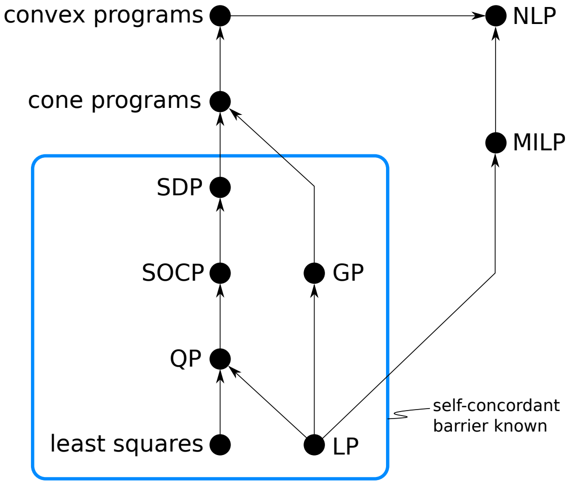
— R. Rockafeller, SIAM Review 1993

Optimization hierarchy



Adapted from Boyd 2002

Optimization hierarchy



Adapted from Boyd 2002

Geometric program: definition

Geometric program: definition

Monomial function

$$m(\boldsymbol{x}) = c\prod_{i=1}^{n} x_i^{a_i}, \qquad c > 0 \qquad\qquad \text{(e.g., } \tfrac{1}{2}\rho V^2 C_L S\text{)}$$

Geometric program: definition

Monomial function

$$m(\boldsymbol{x}) = c\prod_{i=1}^{n} x_i^{a_i}, \qquad c > 0 \qquad \text{(e.g., } \tfrac{1}{2}\rho V^2 C_L S\text{)}$$

Posynomial function: sum of monomials

$$p(\boldsymbol{x}) = \sum_{k=1}^{K} c_k \prod_{i=1}^{n} x_i^{a_{ik}}, \;\; c_k > 0 \quad \text{(e.g., } c_d + \frac{C_L^2}{\pi e A}\text{)}$$

Geometric program: definition

Monomial function

$$m(\boldsymbol{x}) = c\prod_{i=1}^{n} x_i^{a_i}, \quad c > 0 \qquad \text{(e.g., } \tfrac{1}{2}\rho V^2 C_L S\text{)}$$

Posynomial function: sum of monomials

$$p(\boldsymbol{x}) = \sum_{k=1}^{K} c_k \prod_{i=1}^{n} x_i^{a_{ik}}, \;\; c_k > 0 \quad \text{(e.g., } c_d + \frac{C_L^2}{\pi e A}\text{)}$$

Geometric program (GP)

$$\boxed{\begin{array}{lll} \text{minimize} & p_0(\boldsymbol{x}) & \\ \text{subject to} & p_i(\boldsymbol{x}) \le 1, & i = 1, \ldots, N_i, \\ & m_j(\boldsymbol{x}) = 1, & j = 1, \ldots, N_e \end{array}}$$

with p_i posynomial, m_i monomial

$\boldsymbol{x} = (x_1, x_2, \ldots, x_n) > 0$

Geometric program: convex formulation

variable change: $y_i := \log x_i$

Geometric program: convex formulation

variable change: $y_i := \log x_i$

- Monomials $m(\boldsymbol{x}) = c\prod_{i=1}^n x_i^{a_i}$: affine in $\boldsymbol{y}$ after log transform

$$\log m = b + a^T\boldsymbol{y} \qquad (b = \log c)$$

Geometric program: convex formulation

variable change: $y_i := \log x_i$

- Monomials $m(\boldsymbol{x}) = c \prod_{i=1}^{n} x_i^{a_i}$: affine in $\boldsymbol{y}$ after log transform

$$\log m = b + a^T \boldsymbol{y} \qquad (b = \log c)$$

- Posynomials $p(\boldsymbol{x}) = \sum_{k=1}^{K} c_k \prod_{i=1}^{n} x_i^{a_{ik}}$: convex in $\boldsymbol{y}$ after log transform

$$\log p = \log \left(\sum_{k=1}^{K} e^{b_k + a_k^T \boldsymbol{y}} \right)$$

Geometric program: convex formulation

variable change: $y_i := \log x_i$

- Monomials $m(\boldsymbol{x}) = c \prod_{i=1}^n x_i^{a_i}$: affine in $\boldsymbol{y}$ after log transform

$$\log m = b + a^T \boldsymbol{y} \qquad (b = \log c)$$

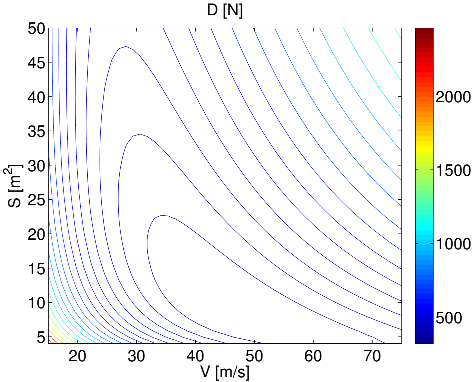
- Posynomials $p(\boldsymbol{x}) = \sum_{k=1}^K c_k \prod_{i=1}^n x_i^{a_{ik}}$: convex in $\boldsymbol{y}$ after log transform

$$\log p = \log \left(\sum_{k=1}^K e^{b_k + a_k^T \boldsymbol{y}} \right)$$

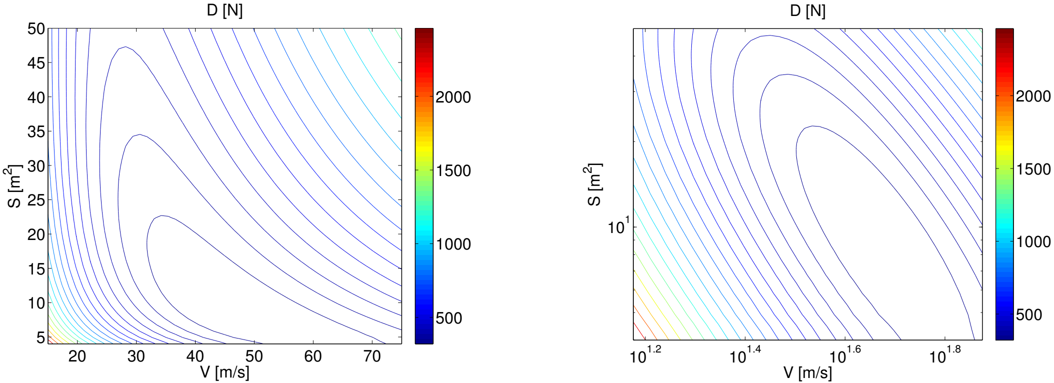
- GP in convex form

$$\boxed{\begin{array}{ll} \text{minimize} & \log \left(\sum_{k=1}^{K_0} \exp(b_{0k} + a_{0k}^T \boldsymbol{y}) \right) \\ \text{subject to} & \log \left(\sum_{k=1}^{K_i} \exp(b_{ik} + a_{ik}^T \boldsymbol{y}) \right) \le 0, \quad i = 1, \ldots, N_p \\ & G\boldsymbol{y} + h = 0 \end{array}}$$

Geometric program: convex formulation

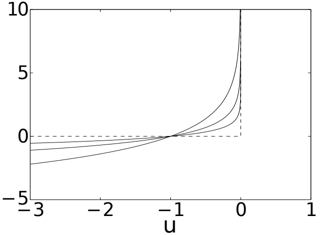


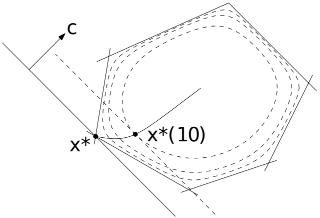
Geometric program: convex formulation



Solving geometric programs

Interior-point methods





Figures: [Boyd 2004]

Benefits:

- Globally optimal solution, guaranteed
- Robust: no initial guesses or parameter tuning
- Off-the-shelf solvers

Boyd GP benchmarks (2005) [1]

- dense GP: 1,000 variables, 10,000 constraints: less than 1 minute
- sparse GP: 10,000 variables, 1,000,000 constraints: "minutes"

Today's talk

Geometric programming overview

Aircraft design modeling examples

Current and future directions

Modeling warm-up: monomial examples

- Steady level flight relations

$$W = \frac{1}{2}\rho V^2 C_L S, \qquad T = \frac{1}{2}\rho V^2 C_D S, \qquad TV = h_{\mathsf{fuel}} \dot{m}_{\mathsf{fuel}} \eta_{\mathsf{thm}} \eta_{\mathsf{eng}} \eta_{\mathsf{prop}}$$

- Non-dimensional coefficients

$$\mathsf{Re} = \frac{\rho V c}{\mu}, \quad M = \frac{V}{\sqrt{\gamma R T}}$$

- Sizing parameters

$$\tau = \frac{t}{c}, \quad \lambda = \frac{c_t}{c_r}, \quad A = \frac{b^2}{S}$$

- Empirical power law models

$$W_{\substack{\mathsf{main}\\ \mathsf{gear}}} = 0.011 W^{0.888} N_{\mathsf{land}}^{0.25} L_{\mathsf{main}}^{0.4} N_{\mathsf{wheel}}^{0.321} N_{\mathsf{ss}}^{-0.5} V_{\mathsf{stall}}^{0.1}$$ [Raymer 2006]

Modeling example 1: Taylor expansion

Breguet range equation

$$R = \frac{h_f}{g}\eta_0\frac{C_L}{C_D}\log(1 + \frac{W_{fuel}}{W_{empty}})$$

Modeling example 1: Taylor expansion

Breguet range equation

$$R = \frac{h_f}{g}\eta_0 \frac{C_L}{C_D} \log(1 + \frac{W_{fuel}}{W_{empty}})$$

$$\exp(\frac{RgC_D}{h_f C_L \eta_0}) = 1 + \frac{W_{fuel}}{W_{empty}}$$

Modeling example 1: Taylor expansion

Breguet range equation

$$R = \frac{h_f}{g}\eta_0\frac{C_L}{C_D}\log(1 + \frac{W_{fuel}}{W_{empty}})$$

$$\exp(\frac{RgC_D}{h_f C_L \eta_0}) = 1 + \frac{W_{fuel}}{W_{empty}}$$

GP formulation

$$z \geq \frac{RgC_D}{h_f C_L \eta_0}$$

$$z + \frac{z^2}{2!} + \frac{z^3}{3!} + ... \leq \frac{W_{fuel}}{W_{empty}}$$

Modeling example 1: Taylor expansion

Breguet range equation

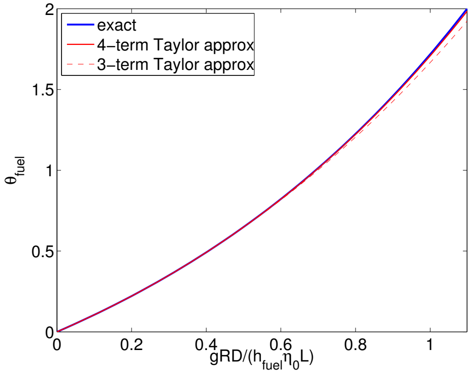
$$R = \frac{h_f}{g}\eta_0 \frac{C_L}{C_D}\log(1 + \frac{W_{fuel}}{W_{empty}})$$

$$\exp(\frac{RgC_D}{h_f C_L \eta_0}) = 1 + \frac{W_{fuel}}{W_{empty}}$$

GP formulation

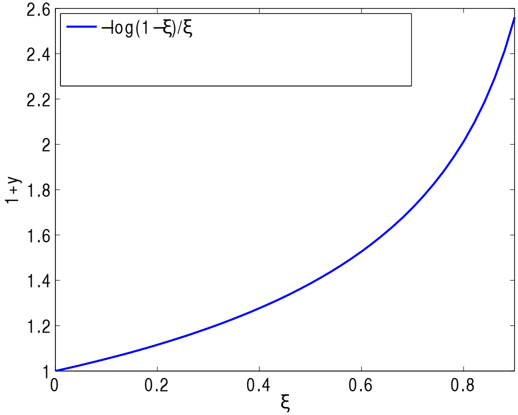
$$z \geq \frac{RgC_D}{h_f C_L \eta_0}$$

$$z + \frac{z^2}{2!} + \frac{z^3}{3!} + \ldots \leq \frac{W_{fuel}}{W_{empty}}$$



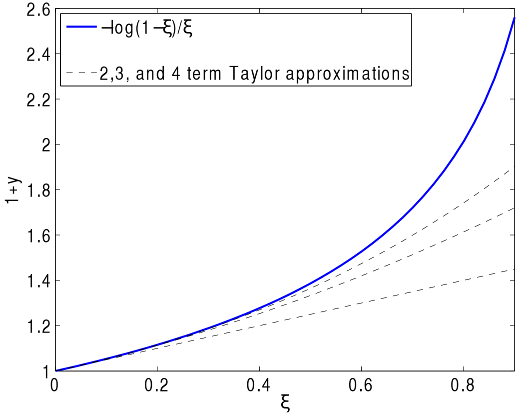
Modeling example 2: Implicit posynomial fitting

$$y = -\frac{\log(1-\xi)}{\xi}$$



Modeling example 2: Implicit posynomial fitting

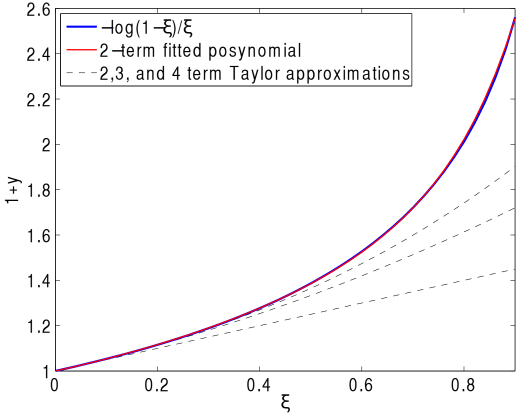
$$y = -\frac{\log(1-\xi)}{\xi}$$



Modeling example 2: Implicit posynomial fitting

$$y = -\frac{\log(1-\xi)}{\xi}$$

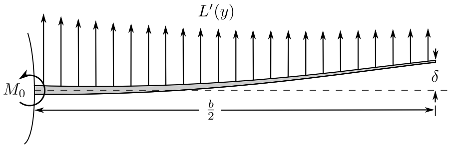
$$1 \geq \frac{0.0464\xi^{2.73}}{y^{2.88}} + \frac{1.044\xi^{0.296}}{y^{0.049}}$$



Modeling example 3: Local function approximation

Stress limit:

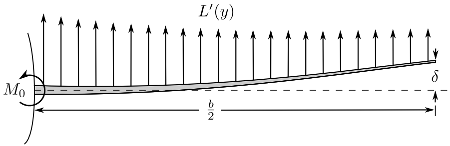
$$\sigma_{safe} S_{root} \geq N_{lift} M_{root}$$

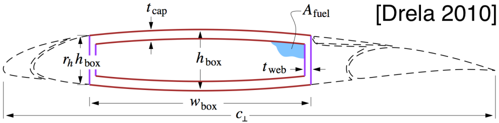


Modeling example 3: Local function approximation

Stress limit:

$$\sigma_{safe} S_{root} \geq N_{lift} M_{root}$$





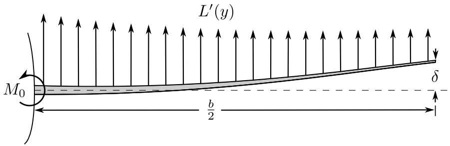
[Drela 2010]

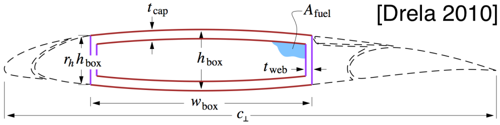
$$\bar{h}_{rms} \bar{w} \bar{t}_{cap}^2 + \bar{I}_{cap} \leq \frac{1}{2} \bar{w} \bar{t}_{cap}$$

Modeling example 3: Local function approximation

Stress limit:

$$\sigma_{safe} S_{root} \geq N_{lift} M_{root}$$



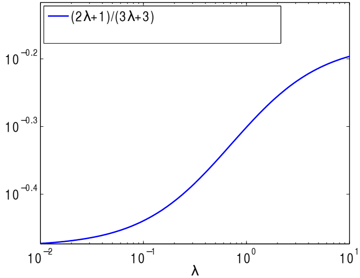


[Drela 2010]

$$\bar{h}_{rms} \bar{w} \bar{t}_{cap}^2 + \bar{I}_{cap} \leq \frac{1}{2} \bar{w} \bar{t}_{cap}$$

Applied root moment:

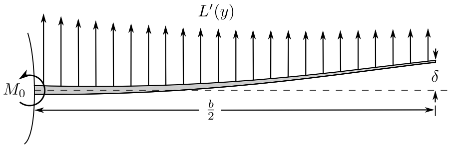
$$M_0 \geq \frac{\tilde{W} b}{4} \left[\frac{2\lambda + 1}{3\lambda + 3} \right]$$

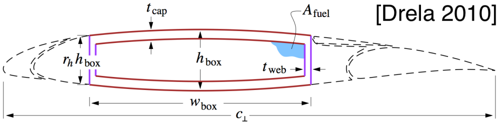


Modeling example 3: Local function approximation

Stress limit:

$$\sigma_{safe} S_{root} \geq N_{lift} M_{root}$$



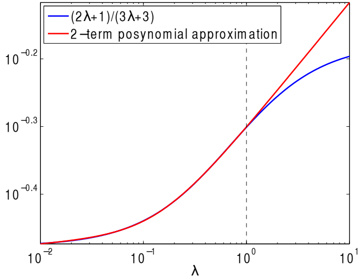


[Drela 2010]

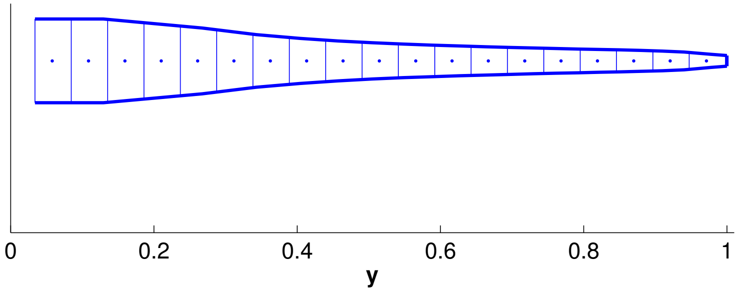
$$\bar{h}_{rms} \bar{w} \bar{t}_{cap}^2 + \bar{I}_{cap} \leq \frac{1}{2} \bar{w} \bar{t}_{cap}$$

Applied root moment:

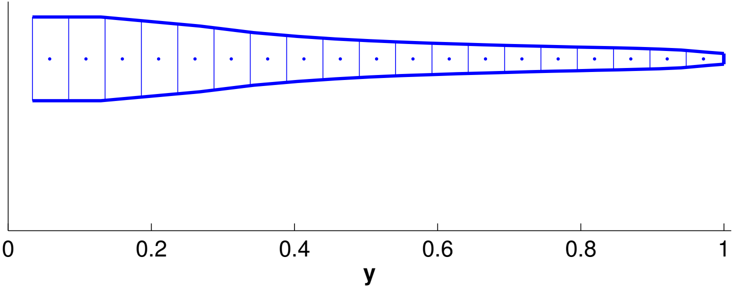
$$M_0 \geq \frac{\tilde{W} b}{4} \left[\frac{2\lambda + 1}{3\lambda + 3} \right]$$



Modeling example 4: spanwise discretization

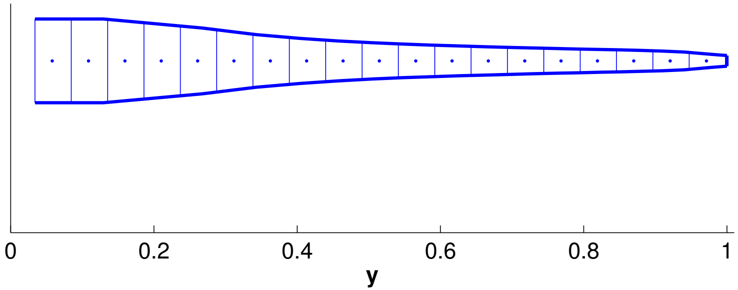


Modeling example 4: spanwise discretization



Problem: $C_P \leq (\Delta C_P)_1 + (\Delta C_P)_2 + \ldots + (\Delta C_P)_N \quad ?$

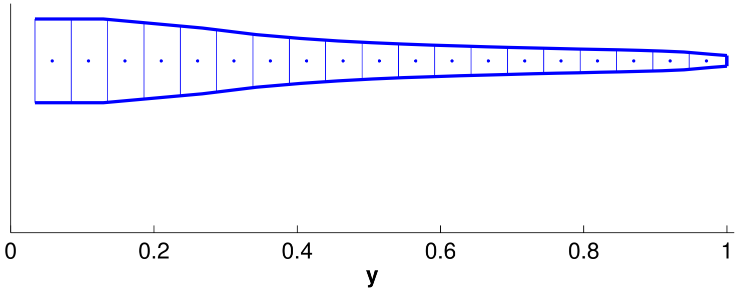
Modeling example 4: spanwise discretization



Problem: $C_P \leq (\Delta C_P)_1 + (\Delta C_P)_2 + \ldots + (\Delta C_P)_N$?

Solution: $C_P \leq N(\Delta C_P)$ (constant power per bin)

Modeling example 4: spanwise discretization



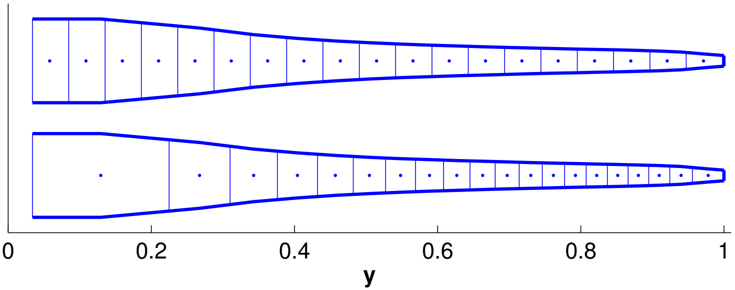
Problem: $C_P \leq (\Delta C_P)_1 + (\Delta C_P)_2 + \ldots + (\Delta C_P)_N \quad ?$

Solution: $C_P \leq N(\Delta C_P)$ (constant power per bin)

$$y_i + \frac{(\Delta y)_i}{2} + \frac{(\Delta y)_{i+1}}{2} \leq y_{i+1}, \;\; i = 1 \ldots (N-1)$$

$$y_N + \frac{(\Delta y)_N}{2} \leq 1$$

Modeling example 4: spanwise discretization



Problem: $C_P \leq (\Delta C_P)_1 + (\Delta C_P)_2 + \ldots + (\Delta C_P)_N$?

Solution: $C_P \leq N(\Delta C_P)$ (constant power per bin)

$$y_i + \frac{(\Delta y)_i}{2} + \frac{(\Delta y)_{i+1}}{2} \leq y_{i+1}, \quad i = 1 \ldots (N-1)$$

$$y_N + \frac{(\Delta y)_N}{2} \leq 1$$

Modeling example 5: Fitting models from data

$\sim 10,000$ data points from

$$c_d(C_L, \mathrm{Re}, \tau)$$

for NACA-24xx airfoils, generated using XFOIL [Drela 1989]

- τ ranging from 8% to 16%
- Re ranging from 10^6 to 10^7 (small homebuilt to small jet)
- C_L ranging from 0 to stall

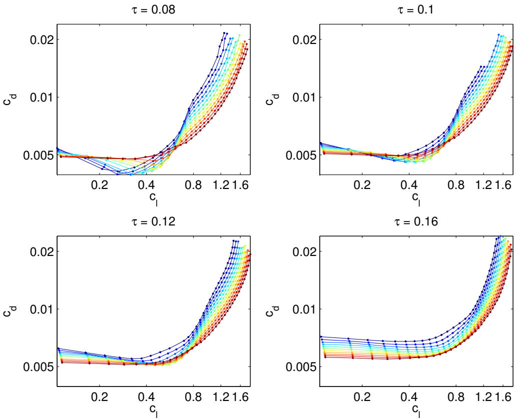
Modeling example 5: Fitting models from data

$\sim 10,000$ data points from

$$c_d(C_L, \mathrm{Re}, \tau)$$

for NACA-24xx airfoils, generated using XFOIL [Drela 1989]

- τ ranging from 8% to 16%
- Re ranging from 10^6 to 10^7 (small homebuilt to small jet)
- C_L ranging from 0 to stall



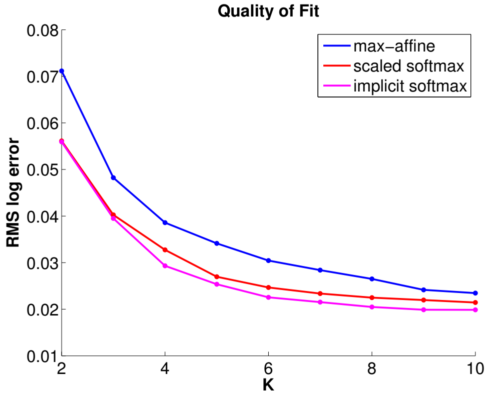
Modeling example 5: Fitting models from data

$\sim 10,000$ data points from

$$c_d(C_L, \mathrm{Re}, \tau)$$

for NACA-24xx airfoils, generated using XFOIL [Drela 1989]

- τ ranging from 8% to 16%
- Re ranging from 10^6 to 10^7 (small homebuilt to small jet)
- C_L ranging from 0 to stall



GPkit: GP modeling in python

- substitution instead of constants

```
gpkit.Variable("R", 8, "meters")
```

- unit checking and conversions

```
gpkit.Variable("W", 4.94, "kilonewtons")
```

- interactive explorations

```
gpkit.interactive.widget(gp)
```

- sweeps over the design space

```
gp.sub("R", ("sweep", [4, 6, 8]))
```

- Experimental: optimization involving random variables

```
gpkit.Variable("\sigma_max", min=220, expected=276, "MPa")
```

http://gpkit.readthedocs.org

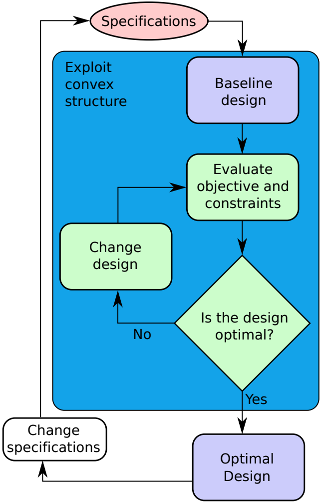
Today's talk

Geometric programming overview

Aircraft design modeling examples

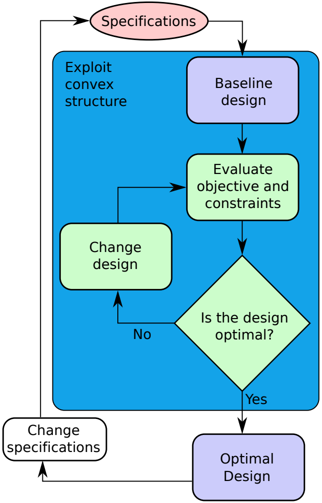
Current and future directions

Constraint sensitivities



Which trade studies should we conduct?

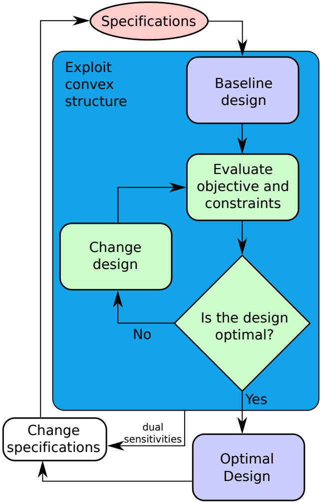
Constraint sensitivities



Which trade studies should we conduct?

- Dual variables quantify sensitivity of objective to each constraint.
- Primal-dual interior point algorithms determine optimal dual variables for free.

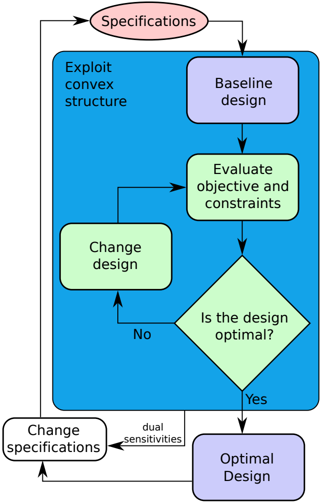
Constraint sensitivities



Which trade studies should we conduct?

- Dual variables quantify sensitivity of objective to each constraint.
- Primal-dual interior point algorithms determine optimal dual variables for free.

Constraint sensitivities

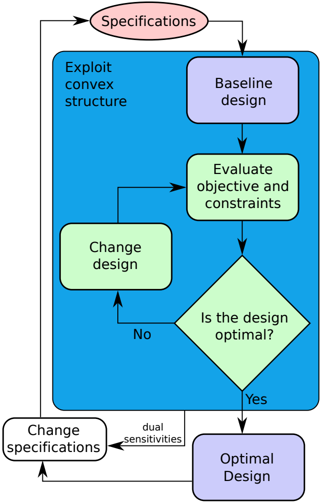


Which trade studies should we conduct?

- Dual variables quantify sensitivity of objective to each constraint.
- Primal-dual interior point algorithms determine optimal dual variables for free.

Example The dual variable associated with constraint i is -0.27.

Constraint sensitivities



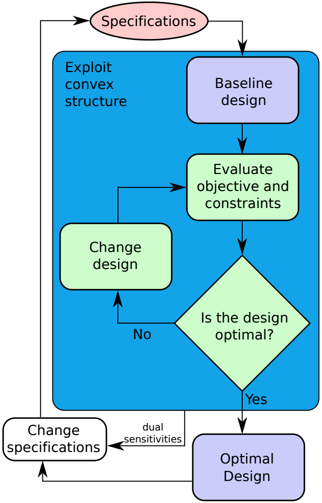
Which trade studies should we conduct?

- Dual variables quantify sensitivity of objective to each constraint.
- Primal-dual interior point algorithms determine optimal dual variables for free.

Example The dual variable associated with constraint i is -0.27.

Interpretation If we *relax* constraint i by 1% and then re-solve, we expect the optimum to improve by 0.27%.

Constraint sensitivities



Which trade studies should we conduct?

- Dual variables quantify sensitivity of objective to each constraint.
- Primal-dual interior point algorithms determine optimal dual variables for free.

Example The dual variable associated with constraint i is -0.27.

Interpretation If we *relax* constraint i by 1% and then re-solve, we expect the optimum to improve by 0.27%.

Applications

- Guide trade studies
- Direct engineering effort to most important areas
- Better understand uncertainty propagation

Feasibility Analysis

When constraints cannot all be satisfied, GP solvers provide a mathematical certificate that no feasible point exists.

Feasibility Analysis

When constraints cannot all be satisfied, GP solvers provide a mathematical certificate that no feasible point exists.

In this case, look for *closest feasible point*:

Original GP

$$\begin{array}{lll} \text{minimize} & p_0(x) & \\ \text{subject to} & p_i(x) \leq 1, & i = 1, \dots, N_p, \\ & m_j(x) = 1, & j = 1, \dots, N_m \end{array}$$

Closest Feasible Point GP

$$\begin{array}{lll} \text{minimize} & s & \\ \text{subject to} & p_i(x) \leq s, & i = 1, \dots, N_p, \\ & m_j(x) = 1, & j = 1, \dots, N_m \end{array}$$

Feasibility Analysis

When constraints cannot all be satisfied, GP solvers provide a mathematical certificate that no feasible point exists.

In this case, look for *closest feasible point*:

Original GP

$$
\begin{array}{lll}
\text{minimize} & p_0(x) & \\
\text{subject to} & p_i(x) \le 1, & i = 1, \ldots, N_p, \\
 & m_j(x) = 1, & j = 1, \ldots, N_m
\end{array}
$$

Closest Feasible Point GP

$$
\begin{array}{lll}
\text{minimize} & s & \\
\text{subject to} & p_i(x) \le s, & i = 1, \ldots, N_p, \\
 & m_j(x) = 1, & j = 1, \ldots, N_m
\end{array}
$$

The closest feasible point GP is always* feasible, and its optimal point is within $100(s-1)$% of satisfying the original inequality constraints.

*assuming monomial equality constraints are feasible

Signomial programming

- Primary limitation of GP approach: models must be log-convex
- Can handle more general models using signomial programming

Signomial programming

$$
\begin{array}{lll}
\text{minimize} & p_0(x) & \\
\text{subject to} & p_i(x) \leq 1, & i = 1, ..., N_p, \\
 & m_j(x) = 1, & j = 1, ..., N_m \\
 & p_k(x) q_k(x) \leq 1 & k = 1, ..., N_s
\end{array}
$$

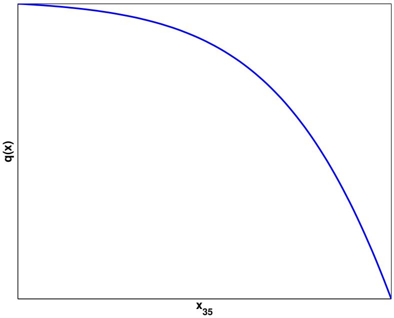
- Primary limitation of GP approach: models must be log-convex
- Can handle more general models using signomial programming

Signomial programming

- Primary limitation of GP approach: models must be log-convex
- Can handle more general models using signomial programming

$$\begin{array}{lll} \text{minimize} & p_0(x) & \\ \text{subject to} & p_i(x) \leq 1, & i = 1, \ldots, N_p, \\ & m_j(x) = 1, & j = 1, \ldots, N_m \\ & p_k(x) q_k(x) \leq 1 & k = 1, \ldots, N_s \end{array}$$

Sketch of sequential GP approach

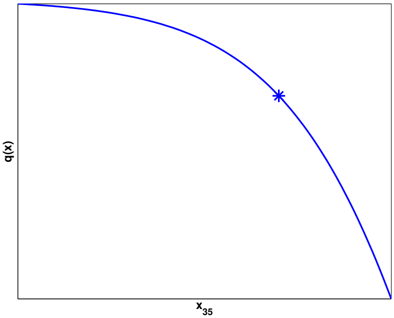


Signomial programming

- Primary limitation of GP approach: models must be log-convex
- Can handle more general models using signomial programming

$$\begin{array}{lll} \text{minimize} & p_0(x) & \\ \text{subject to} & p_i(x) \leq 1, & i = 1, ..., N_p, \\ & m_j(x) = 1, & j = 1, ..., N_m \\ & p_k(x)q_k(x) \leq 1 & k = 1, ..., N_s \end{array}$$

Sketch of sequential GP approach

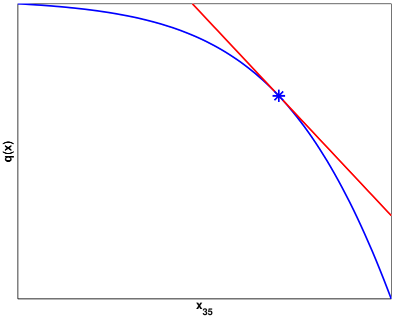


Signomial programming

$$\begin{array}{lll} \text{minimize} & p_0(x) & \\ \text{subject to} & p_i(x) \leq 1, & i = 1, ..., N_p, \\ & m_j(x) = 1, & j = 1, ..., N_m \\ & p_k(x)q_k(x) \leq 1 & k = 1, ..., N_s \end{array}$$

- Primary limitation of GP approach: models must be log-convex
- Can handle more general models using signomial programming

Sketch of sequential GP approach

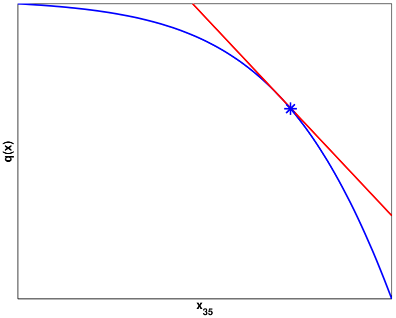


Signomial programming

$$
\begin{array}{lll}
\text{minimize} & p_0(x) & \\
\text{subject to} & p_i(x) \leq 1, & i = 1, \ldots, N_p, \\
& m_j(x) = 1, & j = 1, \ldots, N_m \\
& p_k(x) q_k(x) \leq 1 & k = 1, \ldots, N_s
\end{array}
$$

- Primary limitation of GP approach: models must be log-convex
- Can handle more general models using signomial programming

Sketch of sequential GP approach

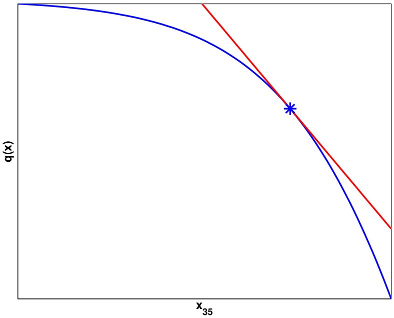


Signomial programming

- Primary limitation of GP approach: models must be log-convex
- Can handle more general models using signomial programming

$$\begin{array}{lll} \text{minimize} & p_0(x) & \\ \text{subject to} & p_i(x) \leq 1, & i = 1, ..., N_p, \\ & m_j(x) = 1, & j = 1, ..., N_m \\ & p_k(x) q_k(x) \leq 1 & k = 1, ..., N_s \end{array}$$

Sketch of sequential GP approach

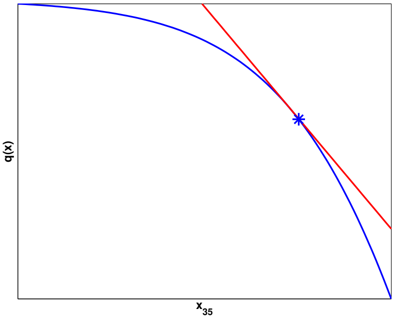


Signomial programming

- Primary limitation of GP approach: models must be log-convex
- Can handle more general models using signomial programming

$$\begin{array}{lll} \text{minimize} & p_0(x) & \\ \text{subject to} & p_i(x) \leq 1, & i = 1, ..., N_p, \\ & m_j(x) = 1, & j = 1, ..., N_m \\ & p_k(x) q_k(x) \leq 1 & k = 1, ..., N_s \end{array}$$

Sketch of sequential GP approach

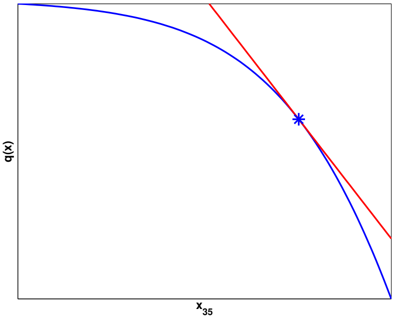


Signomial programming

- Primary limitation of GP approach: models must be log-convex
- Can handle more general models using signomial programming

$$
\begin{array}{lll}
\text{minimize} & p_0(x) & \\
\text{subject to} & p_i(x) \leq 1, & i = 1, \ldots, N_p, \\
& m_j(x) = 1, & j = 1, \ldots, N_m \\
& p_k(x) q_k(x) \leq 1 & k = 1, \ldots, N_s
\end{array}
$$

Sketch of sequential GP approach

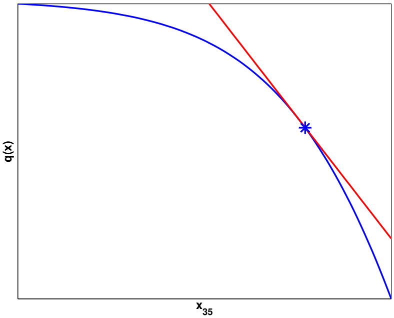


Signomial programming

$$
\begin{array}{lll}
\text{minimize} & p_0(x) & \\
\text{subject to} & p_i(x) \leq 1, & i = 1, ..., N_p, \\
 & m_j(x) = 1, & j = 1, ..., N_m \\
 & p_k(x) q_k(x) \leq 1 & k = 1, ..., N_s
\end{array}
$$

- Primary limitation of GP approach: models must be log-convex
- Can handle more general models using signomial programming

Sketch of sequential GP approach

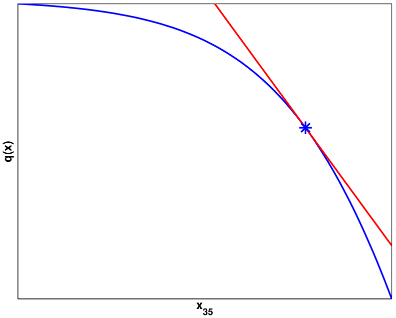


Signomial programming

$$\begin{array}{lll} \text{minimize} & p_0(x) & \\ \text{subject to} & p_i(x) \leq 1, & i = 1, ..., N_p, \\ & m_j(x) = 1, & j = 1, ..., N_m \\ & p_k(x) q_k(x) \leq 1 & k = 1, ..., N_s \end{array}$$

- Primary limitation of GP approach: models must be log-convex
- Can handle more general models using signomial programming

Sketch of sequential GP approach

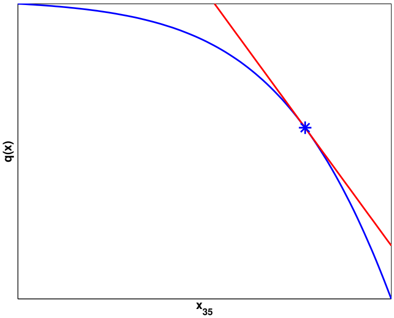


Signomial programming

- Primary limitation of GP approach: models must be log-convex
- Can handle more general models using signomial programming

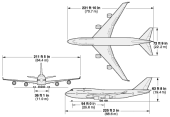
$$
\begin{array}{lll}
\text{minimize} & p_0(x) & \\
\text{subject to} & p_i(x) \le 1, & i = 1, ..., N_p, \\
 & m_j(x) = 1, & j = 1, ..., N_m \\
 & p_k(x) q_k(x) \le 1 & k = 1, ..., N_s
\end{array}
$$

Sketch of sequential GP approach



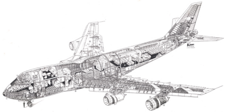
System-level design

Conceptual Design



Preliminary Design

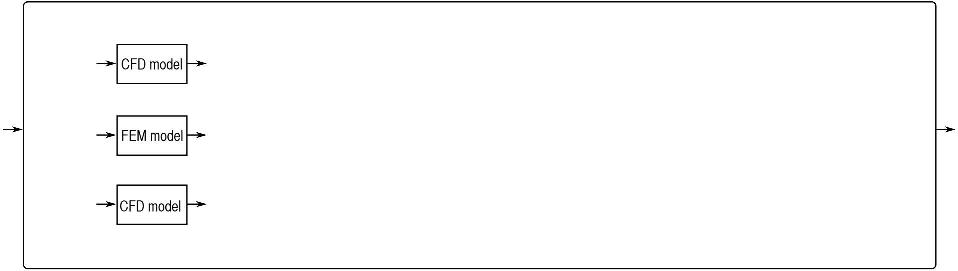
Detailed Design



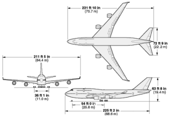
Manufacturing

Operations

System-level design

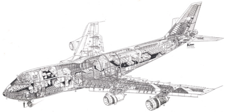


Conceptual Design



Preliminary Design

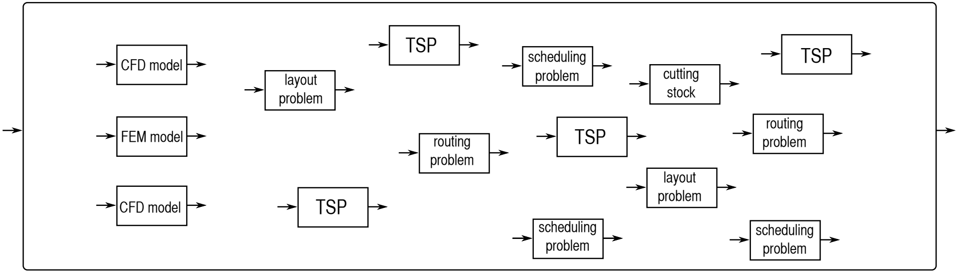
Detailed Design



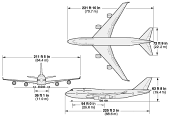
Manufacturing

Operations

System-level design

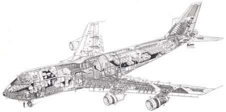


Conceptual Design



Preliminary Design

Detailed Design



Manufacturing

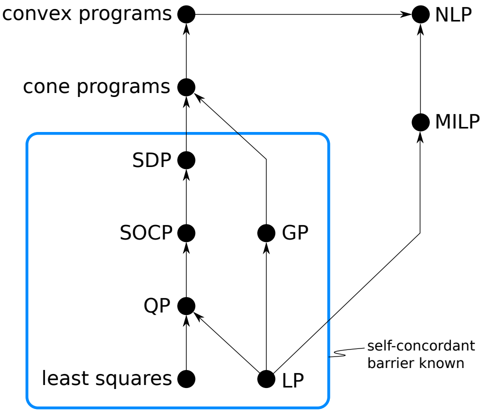
Operations

Take-aways

- Importance of mathematical structure
- Key to tractability: convexity
- Result: reliable and efficient optimization that scales to large problems

Take-aways

- Importance of mathematical structure
- Key to tractability: convexity
- Result: reliable and efficient optimization that scales to large problems

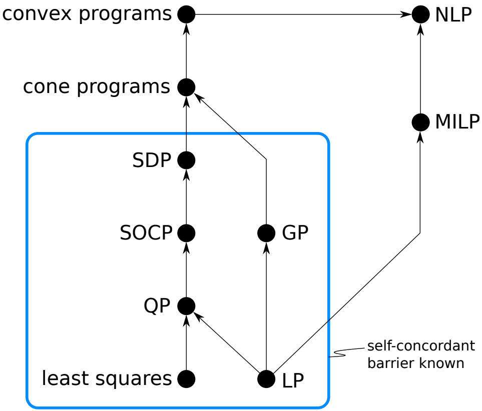


Take-aways

- Importance of mathematical structure
- Key to tractability: convexity
- Result: reliable and efficient optimization that scales to large problems

Current research interests

- Variable transformations for quasi-convex functions
- Signomial programming
- Fitting convex optimization models to data



Acknowledgements

Cody Karcher
Philippe Kirschen
Ned Burnell
Pieter Abbeel
Laurent El Ghaoui
Andrew Packard
Alex Bayen
Karen Willcox
Mark Drela
John Hansman

This work was supported by an NSF Graduate Research Fellowship, a NASA Graduate Research Fellowship, and an MIT Presidential Fellowship.

Questions

References

Stephen Boyd, Seung-Jean Kim, Lieven Vandenberghe, and Arash Hassibi.
A tutorial on geometric programming.
Optimization and Engineering, 2007.

Stephen Boyd and Lieven Vandenberghe.
Convex Optimization.
Cambridge University Press, New York, NY, USA, 2004.

Evin J. Cramer, Jr. J.E. Dennis, Paul D. Frank, Robert Michael Lewis, and Gregory R. Shubin.
Problem formulation for multidisciplinary optimization.
Presented at the AIAA Symposium on Multidisciplinary Design Optimization, August 1993.

T. Bui-Thanh, K. Willcox, and O. Ghattas.
Model reduction for large-scale systems with high-dimensional parametric input space.
SIAM Journal on Scientific Computing, 2008.

Nathan P. Tedford and Joaquim R.R.A. Martins.
Benchmarking multidisciplinary design optimization algorithms.
Optim Eng, February 2009.

John D. Anderson.
Fundamentals of Aerodynamics.
McGraw-Hill, third edition, 2001.

Stephen P. Boyd, Seung jean Kim, Dinesh D. Patil, and Mark A. Horowitz.
Digital circuit optimization via geometric programming.
Operations Research, 53:899–932, 2005.

Warren Hoburg and Pieter Abbeel.
Approximating data with convex functions.
In preparation, 2012.

Mark Drela.
Course notes.

Constraint sensitivities

- Consider perturbed GP:

$$\begin{array}{ll} \text{minimize} & \displaystyle\sum_{k=1}^{K_0} c_{0k} \boldsymbol{x}^{\boldsymbol{a}_{0k}} \\ \text{subject to} & \displaystyle\sum_{k=1}^{K_i} c_{ik} \boldsymbol{x}^{\boldsymbol{a}_{ik}} \le u_i, \;\; i = 1, \ldots, m. \end{array}$$

Constraint sensitivities

- Consider perturbed GP:

$$\begin{array}{ll} \text{minimize} & \displaystyle\sum_{k=1}^{K_0} c_{0k} \boldsymbol{x}^{\boldsymbol{a}_{0k}} \\ \text{subject to} & \displaystyle\sum_{k=1}^{K_i} c_{ik} \boldsymbol{x}^{\boldsymbol{a}_{ik}} \le u_i, \;\; i = 1, ..., m. \end{array}$$

- Define $p^*(\boldsymbol{u}) \equiv$ optimal objective value of perturbed GP

Constraint sensitivities

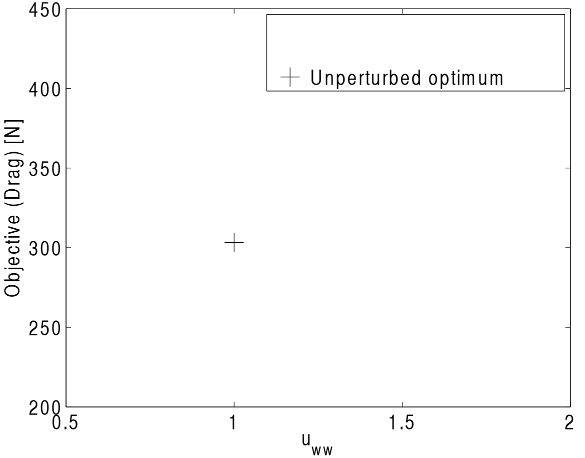
- Consider perturbed GP:

$$\begin{array}{ll} \text{minimize} & \displaystyle\sum_{k=1}^{K_0} c_{0k} \boldsymbol{x}^{\boldsymbol{a}_{0k}} \\ \text{subject to} & \displaystyle\sum_{k=1}^{K_i} c_{ik} \boldsymbol{x}^{\boldsymbol{a}_{ik}} \leq u_i, \;\; i = 1, \ldots, m. \end{array}$$

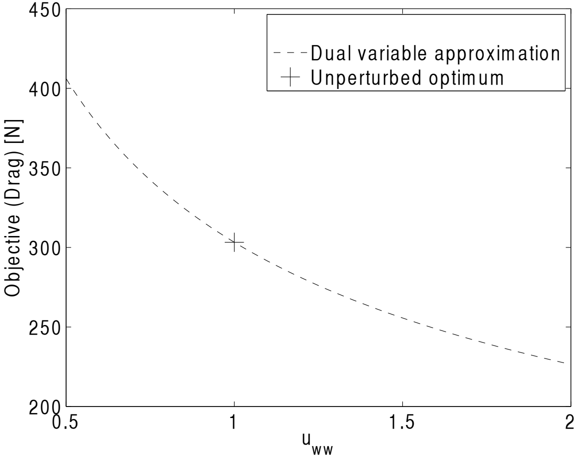
- Define $p^*(\boldsymbol{u}) \equiv$ optimal objective value of perturbed GP

$$\left.\frac{\partial \log p^*(\boldsymbol{u})}{\partial \log u_i}\right|_{\boldsymbol{u}=\boldsymbol{1}} = \left.\frac{\partial \left(\frac{p^*(\boldsymbol{u})}{p^*(\boldsymbol{1})}\right)}{\partial \left(\frac{u_i}{1}\right)}\right|_{\boldsymbol{u}=\boldsymbol{1}} = -\lambda_i$$

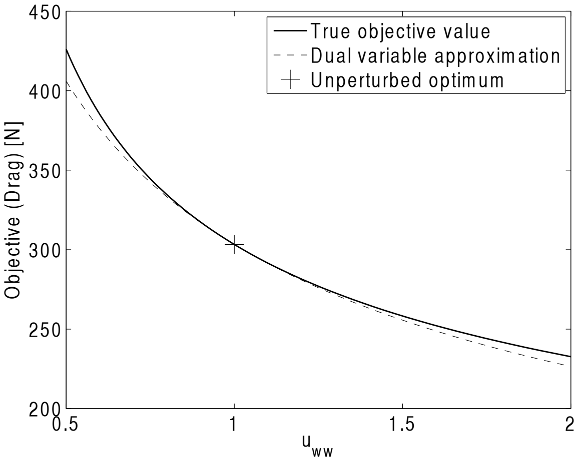
Pareto frontier "linearization"



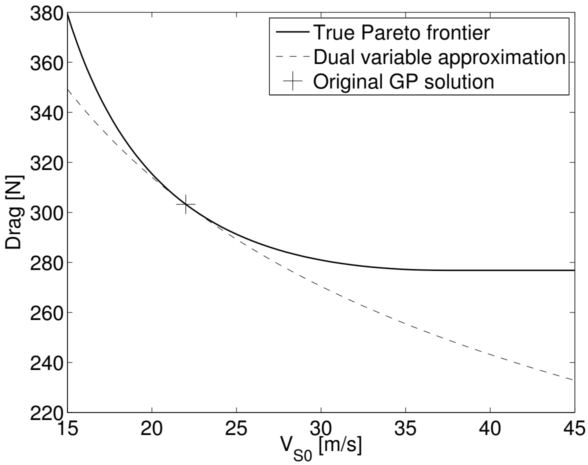
Pareto frontier “linearization”



Pareto frontier "linearization"



Pareto frontier "linearization"



Some Useful Bounds

Dual Sensitivity Analysis

- Perturb constraints (via $\boldsymbol{u}$)
- Performance bound:

$$\log p^*(\boldsymbol{u}) \geq \log p^*(\mathbf{1}) + \boldsymbol{\lambda}^T \boldsymbol{u}$$

- An *optimistic* estimate

Some Useful Bounds

Dual Sensitivity Analysis

- Perturb constraints (via $\boldsymbol{u}$)
- Performance bound:

$$\log p^*(\boldsymbol{u}) \geq \log p^*(\mathbf{1}) + \boldsymbol{\lambda}^T \boldsymbol{u}$$

- An *optimistic* estimate

Design Averaging

- Consider two designs θ_1, θ_2, with objective values p_1^*, p_2^*
- Form geometric mean design

$$\theta_3^{(i)} = \sqrt{\theta_1^{(i)} \theta_2^{(i)}}$$

- Performance bound:

$$p_3^* \leq \sqrt{p_1^* p_2^*}$$

- A *pessimistic* estimate

Posynomial Equality Relaxation

When can we guarantee $h(x) = 1$ will hold at optimum ?

$$
\begin{array}{lll}
\text{minimize} & p_0(x) & \\
\text{subject to} & p_i(x) \leq 1, & i = 1, \ldots, N_p, \\
 & m_j(x) = 1, & j = 1, \ldots, N_m \\
 & h(x) \leq 1 &
\end{array}
$$

Posynomial Equality Relaxation

When can we guarantee $h(x) = 1$ will hold at optimum ?

$$
\begin{array}{lll}
\text{minimize} & p_0(x) & \\
\text{subject to} & p_i(x) \leq 1, & i = 1, ..., N_p, \\
 & m_j(x) = 1, & j = 1, ..., N_m \\
 & h(x) \leq 1 &
\end{array}
$$

If $\exists\ x_k$ s.t.:

- x_k does not appear in monomial equality constraints, i.e. $\frac{\partial m_j}{\partial x_k} = 0$
- p_0 monotone strictly decreasing in x_k, i.e. $\frac{\partial p_0}{\partial x_k} < 0$
- All p_i monotone decreasing in x_k, i.e. $\frac{\partial p_i}{\partial x_k} \leq 0$
- h is monotone strictly increasing in x_k, i.e. $\frac{\partial h}{\partial x_k} > 0$

$\rightarrow$ **Conditions satisfied for all relaxations presented today.**

Extensions exist for multiple $h_i(x)$, $\frac{\partial p_0}{\partial x_k} = 0$ case [Boyd et. al., 2007]

Conceptual Design – Modeling Summary

- Fuselage Pressure Loads
- Fuselage Bending Loads
- Fuselage Weight
- Steady Level Flight Relations
- Wing Moments and Stresses
- Wing Weight
- Stability
- Tail Moments and Stresses
- Tail Weight
- Engine Weight
- Turbine Cycle Analysis
- Noise
- CG Envelope
- Active Gust Response
- Wing Profile Drag
- V-speeds and critical loading cases
- Wing Induced Drag
- Tail Drag
- Fuselage Drag
- Interference Drags
- Airfoil Shape Optimization
- Laminar Flow Control
- Compressibility Effects
- Propulsive Efficiency
- Blade Element Momentum Theory
- APU Sizing
- Hydraulic, Fuel, & Electrical System Weights
- Mission Breakdown and Fuel Burn
- Cruise Climb
- Loiter Performance/Endurance
- Takeoff Distance & 50' obstacle clearance
- Landing Distance
- Spoiler Sizing
- Climb Performance
- Engine-Out Operation
- Windmilling Drag
- Maneuverability
- High Lift System Sizing
- Control Surface Sizing
- Landing Gear Sizing
- Engine Ground Clearance
- Tail Strike Clearance
- Maintenance Costs
- Material Costs
- Manufacturability
- Assembly/Integration Time and Cost
- Fastener Count
- Supply Chain Dynamics

Conceptual Design – Modeling Summary

- Fuselage Pressure Loads
- Fuselage Bending Loads
- Fuselage Weight
- Steady Level Flight Relations
- Wing Moments and Stresses
- Wing Weight
- Stability
- Tail Moments and Stresses
- Tail Weight
- Engine Weight
- Turbine Cycle Analysis
- Noise
- CG Envelope
- Active Gust Response
- Wing Profile Drag
- V-speeds and critical loading cases
- Wing Induced Drag
- Tail Drag
- Fuselage Drag
- Interference Drags
- Airfoil Shape Optimization
- Laminar Flow Control
- Compressibility Effects
- Propulsive Efficiency
- Blade Element Momentum Theory
- APU Sizing
- Hydraulic, Fuel, & Electrical System Weights
- Mission Breakdown and Fuel Burn
- Cruise Climb
- Loiter Performance/Endurance
- Takeoff Distance & 50' obstacle clearance
- Landing Distance
- Spoiler Sizing
- Climb Performance
- Engine-Out Operation
- Windmilling Drag
- Maneuverability
- High Lift System Sizing
- Control Surface Sizing
- Landing Gear Sizing
- Engine Ground Clearance
- Tail Strike Clearance
- Maintenance Costs
- Material Costs
- Manufacturability
- Assembly/Integration Time and Cost
- Fastener Count
- Supply Chain Dynamics

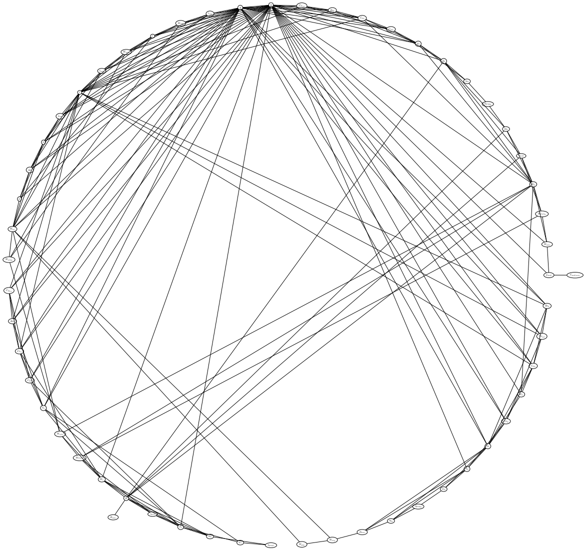
Conceptual Design – Modeling Summary

- Fuselage Pressure Loads
- Fuselage Bending Loads
- Fuselage Weight
- Steady Level Flight Relations
- Wing Moments and Stresses
- Wing Weight
- Stability
- Tail Moments and Stresses
- Tail Weight
- Engine Weight
- Turbine Cycle Analysis
- Noise
- CG Envelope
- Active Gust Response
- Wing Profile Drag
- V-speeds and critical loading cases
- Wing Induced Drag
- Tail Drag
- Fuselage Drag
- Interference Drags
- Airfoil Shape Optimization
- Laminar Flow Control
- Compressibility Effects
- Propulsive Efficiency
- Blade Element Momentum Theory
- APU Sizing
- Hydraulic, Fuel, & Electrical System Weights
- Mission Breakdown and Fuel Burn
- Cruise Climb
- Loiter Performance/Endurance
- Takeoff Distance & 50' obstacle clearance
- Landing Distance
- Spoiler Sizing
- Climb Performance
- Engine-Out Operation
- Windmilling Drag
- Maneuverability
- High Lift System Sizing
- Control Surface Sizing
- Landing Gear Sizing
- Engine Ground Clearance
- Tail Strike Clearance
- Maintenance Costs
- Material Costs
- Manufacturability
- Assembly/Integration Time and Cost
- Fastener Count
- Supply Chain Dynamics

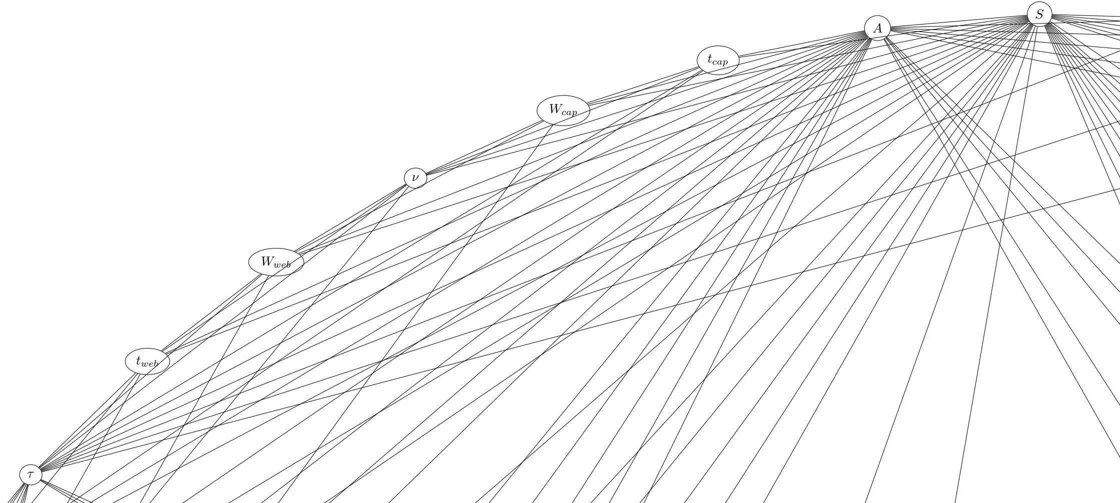
gpkit

- Open source python modeling tool
- Interfaces with MOSEK and cvxopt solvers
- Stable version release planned for November 2014
- `http://github.com/appliedopt/gpkit`

gpkit — coupling graphs



gpkit — coupling graphs



Lagrange Dual of GP

Primal problem (in convex form):

$$\begin{aligned} \text{minimize} \quad & \log \sum_{k=1}^{K_0} \exp(\boldsymbol{a}_{0k}^T \boldsymbol{y} + b_{0k}) \\ \text{subject to} \quad & \log \sum_{k=1}^{K_i} \exp(\boldsymbol{a}_{ik}^T \boldsymbol{y} + b_{ik}) \leq 0, \;\; i = 1, ..., m, \end{aligned} \tag{1}$$

Lagrange Dual of GP

Primal problem (in convex form):

$$\begin{array}{ll} \text{minimize} & \log \sum_{k=1}^{K_0} \exp(\boldsymbol{a}_{0k}^T \boldsymbol{y} + b_{0k}) \\ \text{subject to} & \log \sum_{k=1}^{K_i} \exp(\boldsymbol{a}_{ik}^T \boldsymbol{y} + b_{ik}) \leq 0, \;\; i = 1, ..., m, \end{array} \tag{1}$$

Lagrangian and dual function:

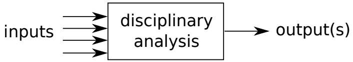
$$\begin{aligned} L(\boldsymbol{y}, \boldsymbol{z}, \boldsymbol{\lambda}, \boldsymbol{\nu}) &= \log \sum_{k=1}^{K_0} \exp z_{0k} + \sum_{i=1}^{m} \lambda_i \log \sum_{k=1}^{K_i} \exp z_{ik} + \sum_{i=0}^{m} \boldsymbol{\nu}_i^T (\boldsymbol{A}_i \boldsymbol{y} + \boldsymbol{b}_i - \boldsymbol{z}_i) \\ g(\boldsymbol{\lambda}, \boldsymbol{\nu}) &= \inf_{\boldsymbol{y}, \boldsymbol{z}} L(\boldsymbol{y}, \boldsymbol{z}, \boldsymbol{\lambda}, \boldsymbol{\nu}). \end{aligned}$$

Lagrange Dual of GP

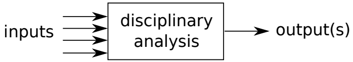
$$
\begin{array}{ll}
\text{maximize} & \displaystyle\sum_{i=0}^{m}\left[\boldsymbol{\nu}_i^T \boldsymbol{b}_i - \sum_{k=1}^{K_i} \nu_{ik} \log \frac{\nu_{ik}}{\mathbf{1}^T \boldsymbol{\nu}_i}\right] \\
\text{subject to} & \displaystyle\sum_{i=0}^{m} \boldsymbol{\nu}_i^T \boldsymbol{A}_i = 0 \\
& \boldsymbol{\nu}_i \geq 0, \quad i = 0, \dots, m \\
& \mathbf{1}^T \boldsymbol{\nu}_0 = 1.
\end{array}
$$

- An equality-constrained entropy maximization
- (unnormalized) probability distributions $\boldsymbol{\nu}_i$ satisfy $\mathbf{1}^T \boldsymbol{\nu}_i = \lambda_i$

Fitting Reduced-Order GP-compatible Models

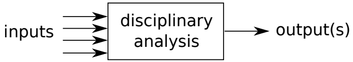


Fitting Reduced-Order GP-compatible Models



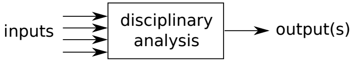
- GP-compatible models can approximate *any* log-convex data [Boyd 2007]

Fitting Reduced-Order GP-compatible Models

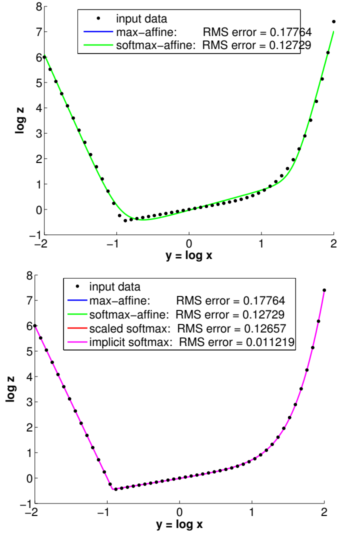


- GP-compatible models can approximate *any* log-convex data [Boyd 2007]
- Given set of data points $(x_1, y_1), \ldots, (x_m, y_m) \in \mathbb{R}^n \times \mathbb{R}$

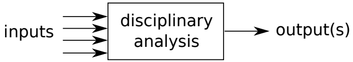
Fitting Reduced-Order GP-compatible Models

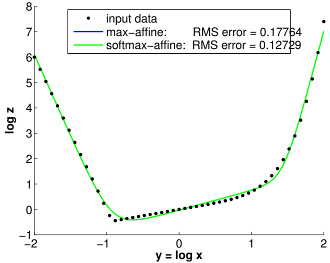


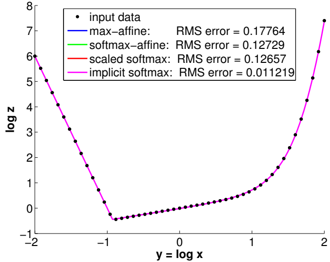
- GP-compatible models can approximate *any* log-convex data [Boyd 2007]
- Given set of data points $(x_1, y_1), \ldots, (x_m, y_m) \in \mathbb{R}^n \times \mathbb{R}$
- Minimize fitting error $||y - f(x)||$, subject to $f \in \mathcal{F}$



Fitting Reduced-Order GP-compatible Models

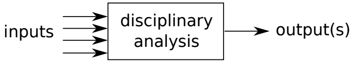




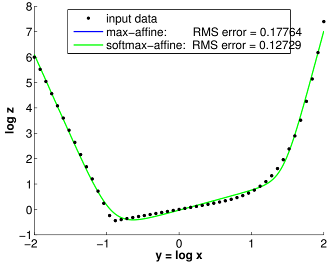


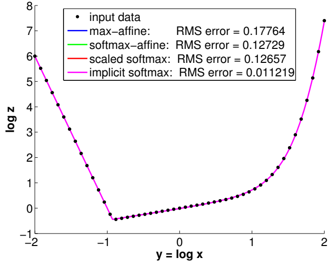
- GP-compatible models can approximate *any* log-convex data [Boyd 2007]
- Given set of data points $(x_1, y_1), \ldots, (x_m, y_m) \in \mathbb{R}^n \times \mathbb{R}$
- Minimize fitting error $||y - f(x)||$, subject to $f \in \mathcal{F}$
- Several choices for $\mathcal{F}$, e.g.
  - Max-affine functions
  - Softmax affine functions
  - Implicit posynomials

Fitting Reduced-Order GP-compatible Models

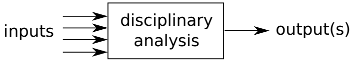


- GP-compatible models can approximate *any* log-convex data [Boyd 2007]
- Given set of data points $(x_1, y_1), \ldots, (x_m, y_m) \in \mathbb{R}^n \times \mathbb{R}$
- Minimize fitting error $||y - f(x)||$, subject to $f \in \mathcal{F}$
- Several choices for $\mathcal{F}$, e.g.
  - Max-affine functions
  - Softmax affine functions
  - Implicit posynomials
- Fitting problem solved offline using trust region Newton methods





Fitting Reduced-Order GP-compatible Models



- GP-compatible models can approximate *any* log-convex data [Boyd 2007]
- Given set of data points $(x_1, y_1), \ldots, (x_m, y_m) \in \mathbb{R}^n \times \mathbb{R}$
- Minimize fitting error $||y - f(x)||$, subject to $f \in \mathcal{F}$
- Several choices for $\mathcal{F}$, e.g.
  - Max-affine functions
  - Softmax affine functions
  - Implicit posynomials
- Fitting problem solved offline using trust region Newton methods
- Many extensions, e.g. conservative fitting, sparse fitting