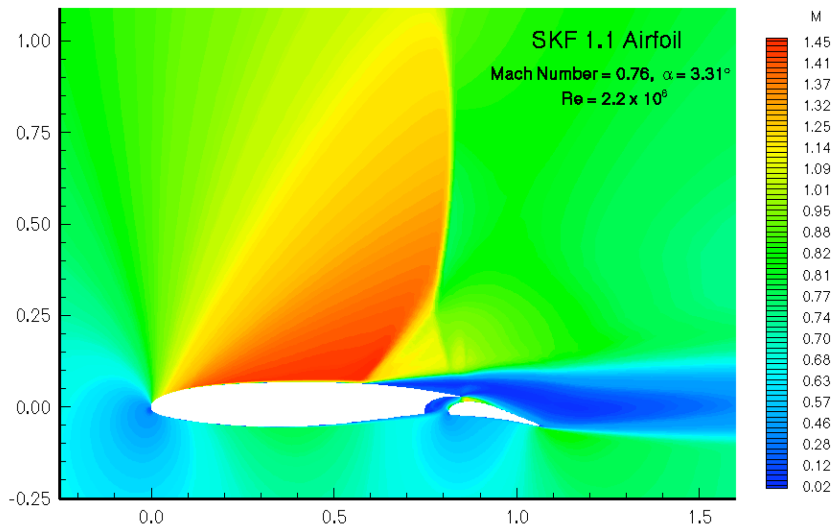
# TOPICS IN AERODYNAMIC SHAPE OPTIMIZATION




Professor David W. Zingg
Canada Research Chair in Computational Aerodynamics
2004 Guggenheim Fellow

University of Toronto
Institute for Aerospace Studies

# BIG PICTURE

- GLOBAL WARMING
  - Air travel is the fastest growing source of greenhouse gas emissions
  - World passenger air traffic increased by about 14% in 2004
  - The world aircraft fleet is expected to double by 2020
- A SOLUTION?
  - Radical new ultra-low-drag aircraft concepts
  - Active flow control
  - Adaptive and morphing wings
- AERODYNAMIC SHAPE OPTIMIZATION
  - Needed for development and evaluation of new concepts

# OUTLINE

- A NEWTON-KRYLOV APPROACH TO AERODYNAMIC OPTIMIZATION
  - a brief description
- TOPIC 1: COMPARISON WITH A GENETIC ALGORITHM
  - examine quantitatively the dependence of the relative speed of the two algorithms on the degree of convergence needed and the number of design variables
- TOPIC 2: OPTIMIZATION UNDER VARIABLE OPERATING CONDITIONS
  - issues in problem formulation
  - an automated approach

# AERODYNAMIC SHAPE OPTIMIZATION

- much faster than traditional "cut"–and–try approach
- more likely to achieve a truly optimal design
- provides insight into the nature of the design space and the trade-offs between various objectives and operating points
- requires that the design problem be completely and carefully specified
- particularly beneficial for new configurations and concepts
- essential for rapid evaluation of competing concepts

# Optimization Problem

**design variables** $X \longrightarrow$ shape of airfoil, $\alpha$, gap, overlap . . .

**state variables** $Q \longrightarrow$ density, momentum, energy . . .

**objective or cost functional** $\mathcal{J}\left[X, Q(X)\right]$
$\longrightarrow$ inverse design, drag, lift, moment, . . .
$\longrightarrow$ multi-point and multi-objective design problems

**constraints** $\longrightarrow$ geometry constraints, such as thickness or area
$\longrightarrow$ flow equations and boundary conditions: $R\left[X, Q(X)\right] = 0$
$\longrightarrow$ flow constraints, such as $C_{\mathrm{p}}$ and $C_{\mathrm{f}}$

Minimize $\mathcal{J}$, subject to satisfying $R = 0$
and any other side constraints.

- **Gradient-based algorithms**
  - various options: constrained/unconstrained, steepest descent, SQP, quasi-Newton, BFGS update
  - line search along gradient direction
  - cost and difficulty of computing the gradient
    - finite difference: one nonlinear solve per design variable per iteration
    - flow sensitivities: one linear solve per design variable per iteration
    - adjoint method: one linear solve per objective or constraint *independent of the number of design variables*
  - adjoint approach is not a "black box": high development cost
  - local optimum
  - difficulties with topology changes, noisy design spaces, inaccurate gradients, categorical variables

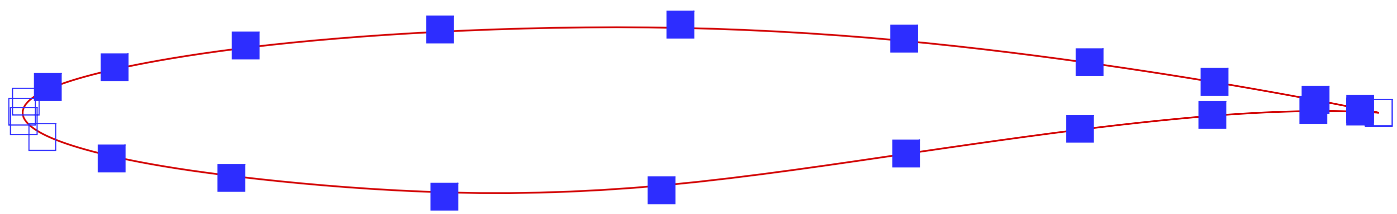
- **Evolutionary (genetic) and search algorithms**
  - global optimum
  - "black box": solver independent
  - slow to converge
  - convergence criteria?
  - aided by response surface/surrogate

# The Newton-Krylov Approach

- Newton-Krylov flow solver for the compressible Navier-Stokes equations with a one-equation turbulence model
  - inexact-Newton strategy
  - matrix-free Krylov method (GMRES)
  - ILU(p) preconditioning based on reduced-storage Jacobian
- discrete adjoint gradient computation
  - ILU-preconditioned Krylov method (matrix-free not possible)
- quasi-Newton method for unconstrained optimization
  - constraints added to objective function as penalty terms
  - BFGS update for approximate inverse Hessian
  - line search based on cubic interpolation

# GEOMETRY PARAMETERIZATION

- cubic B-spline curves
- B-spline control points are the design variables
- angle of incidence provides an additional design variable



Example: 24 control points, 5 frozen: 19 design variables plus $\alpha$

# GENETIC ALGORITHM

- Holst & Pulliam, NASA, 2001
- combination of ranking and selection techniques, mutations, and perturbations
- several parameters: number of chromosomes in a generation (population size), probabilities of selection, mutation, and cross-over
- parallel implementation
- Pareto fronts computed either by the weighted-sum approach or by the dominance Pareto front technique

# TOPIC I: COMPARATIVE EVALUATION: GRADIENT-BASED VS. GENETIC ALGORITHM

- identical flow solver and solver parameters
- identical geometry parameterization
- identical initial and modified meshes
- identical objective functions and geometric constraints
- identical design spaces
- cost measured in terms of (equivalent) objective function evaluations, i.e. flow solutions

# COMPARATIVE EVALUATION: GRADIENT-BASED VS. GENETIC ALGORITHM

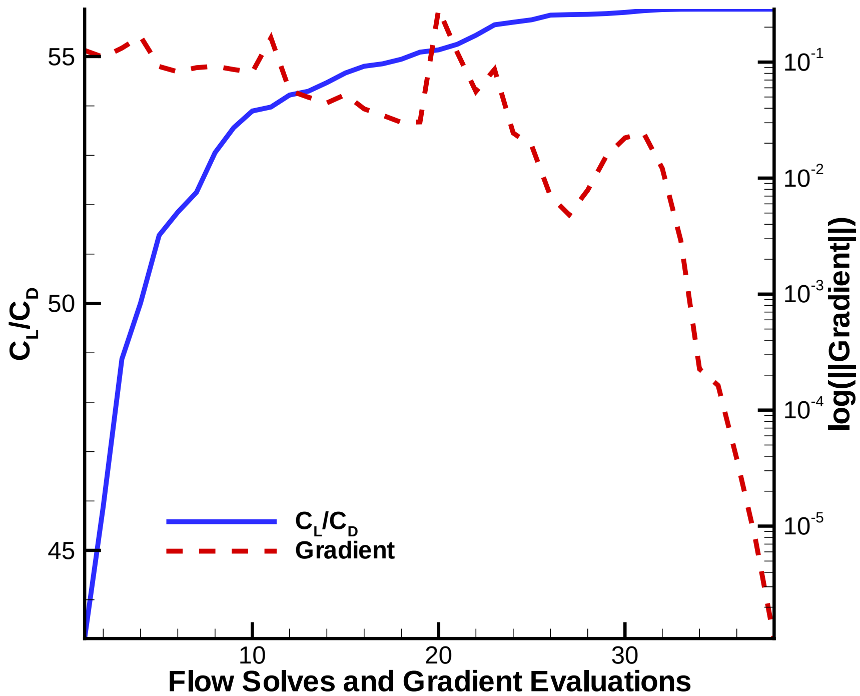
- cost measured in terms of (equivalent) objective function evaluations, i.e. flow solutions
  - genetic algorithm cost roughly equal the product of the population size and the number of generations
- one gradient computation = one flow solution
  - Newton-Krylov algorithm cost equal to twice the number of iterations

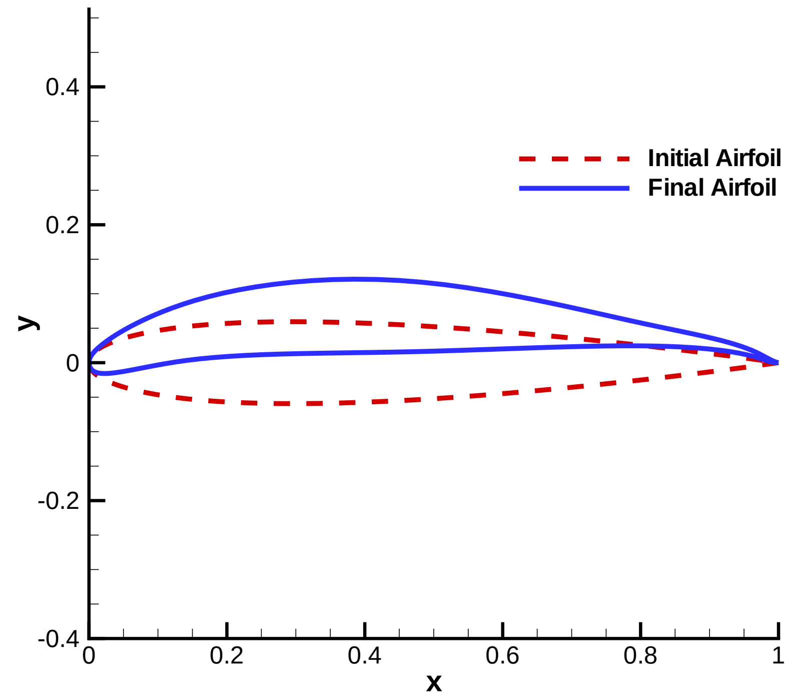
# Problems 1–3: Single-Point Optimization

- Objective: maximize lift-to-drag ratio
- Initial airfoil shape: NACA 0012
- Operating conditions: $M_\infty = 0.25$, $\mathrm{Re} = 2.88$ million
- Geometric constraints: six thickness constraints
- Design variables: angle of incidence plus 8/18/34 B-spline control point locations

| constraint number | 1 | 2 | 3 | 4 | 5 | 6 |
|---|---|---|---|---|---|---|
| location ($\%c$) | 5.0 | 35.0 | 65.0 | 85.0 | 95.0 | 99.0 |
| thickness ($\%c$) | 4.0 | 11.0 | 4.0 | 2.6 | 1.2 | 0.2 |

# Lift-to-Drag Ratio Maximization with 9 Design Variables




Time required: about 45 minutes

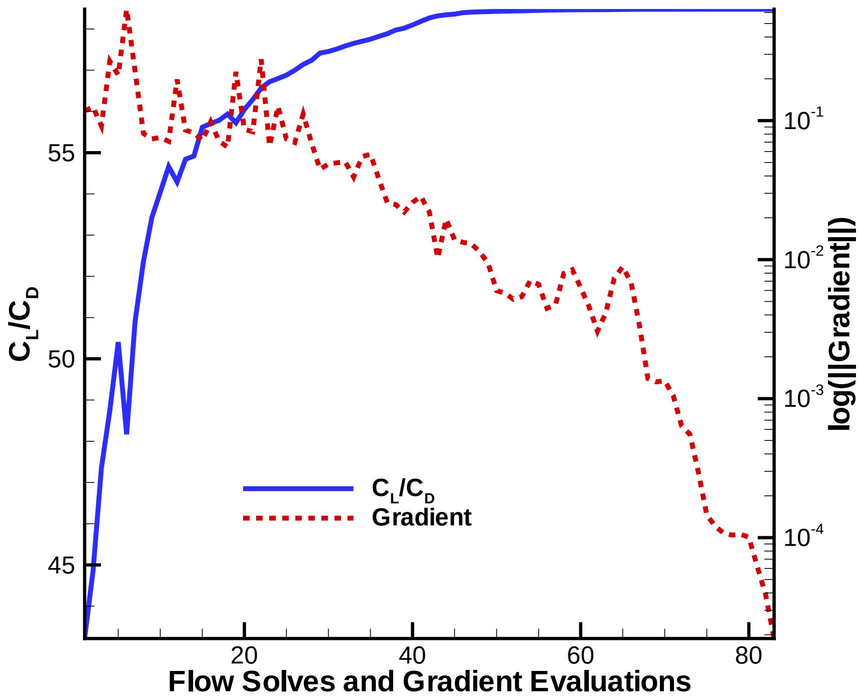
Cost: 76 equivalent function evaluations

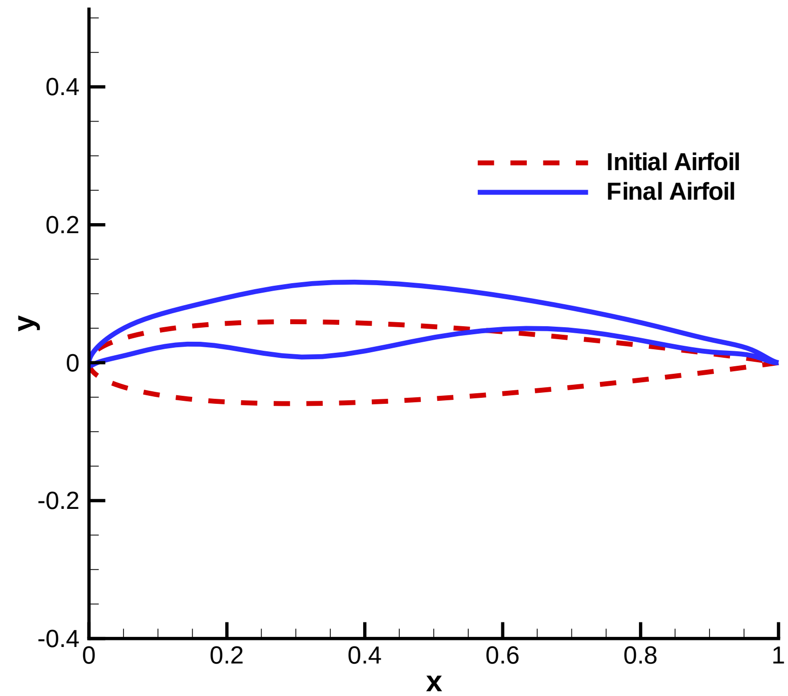
# Lift-to-Drag Ratio Maximization with 9 Design Variables

| % | gradient-based algorithm | genetic algorithm | ratio |
|---|---|---|---|
| 90 | 32 | 440 | 14 |
| 95 | 44 | 951 | 22 |
| 98 | 50 | 2307 | 46 |
| 99 | 52 | 6190 | 119 |

Cost in Equivalent Function Evaluations

# Lift-to-Drag Ratio Maximization with 19 Design Variables




Time required: about 100 minutes

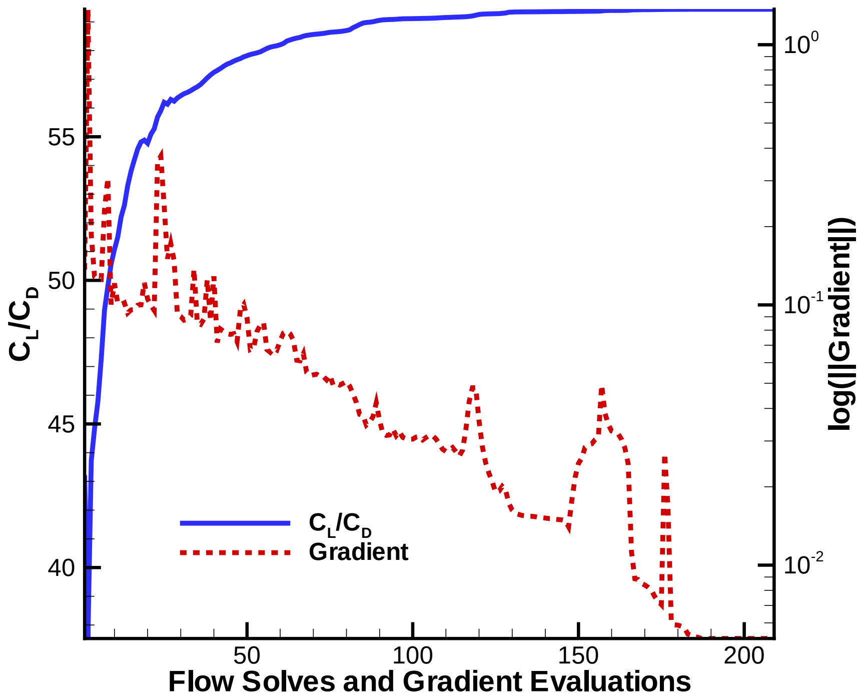
Cost: 166 equivalent function evaluations

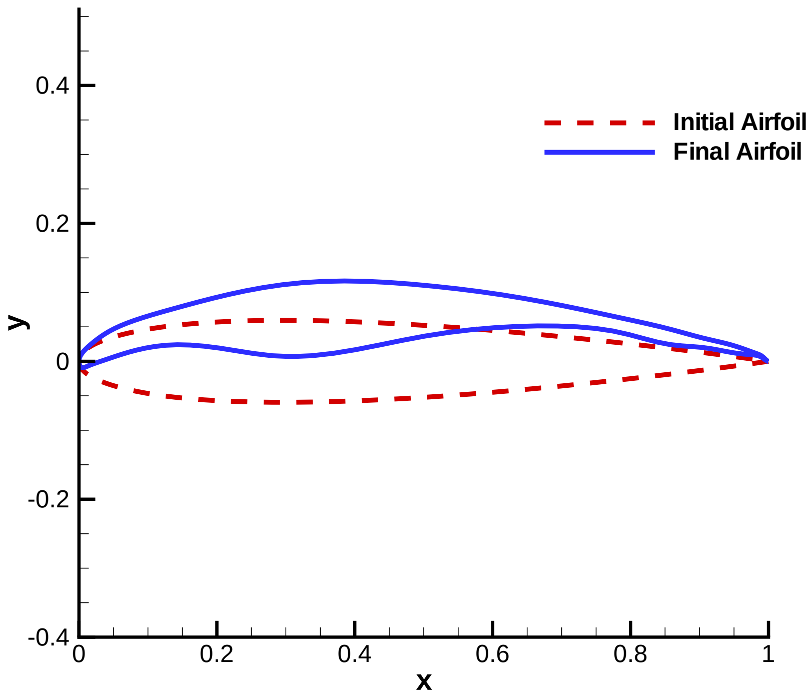
# Lift-to-Drag Ratio Maximization with 19 Design Variables

| % | gradient-based algorithm | genetic algorithm | ratio |
|---|---|---|---|
| 90 | 52 | 3502 | 67 |
| 95 | 70 | 5555 | 79 |
| 98 | 82 | 7833 | 96 |
| 99 | 88 | ???? | ??? |

Cost in Equivalent Function Evaluations

# Lift-to-Drag Ratio Maximization with 35 Design Variables




Time required: about 210 minutes

Cost: 360 equivalent function evaluations

# Lift-to-Drag Ratio Maximization with 35 Design Variables

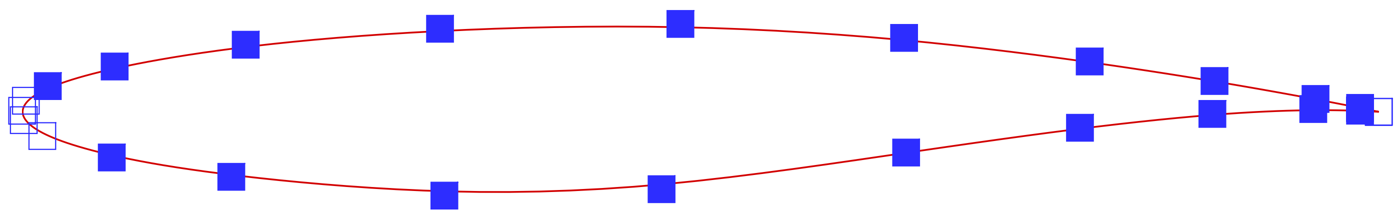
| % | gradient-based algorithm | genetic algorithm | ratio |
|---|---|---|---|
| 90 | 100 | ???? | ?? |
| 95 | 150 | ???? | ?? |
| 98 | 202 | ???? | ?? |
| 99 | 246 | ???? | ??? |

Cost in Equivalent Function Evaluations

# Lift-Constrained Drag Minimization at Transonic Speed

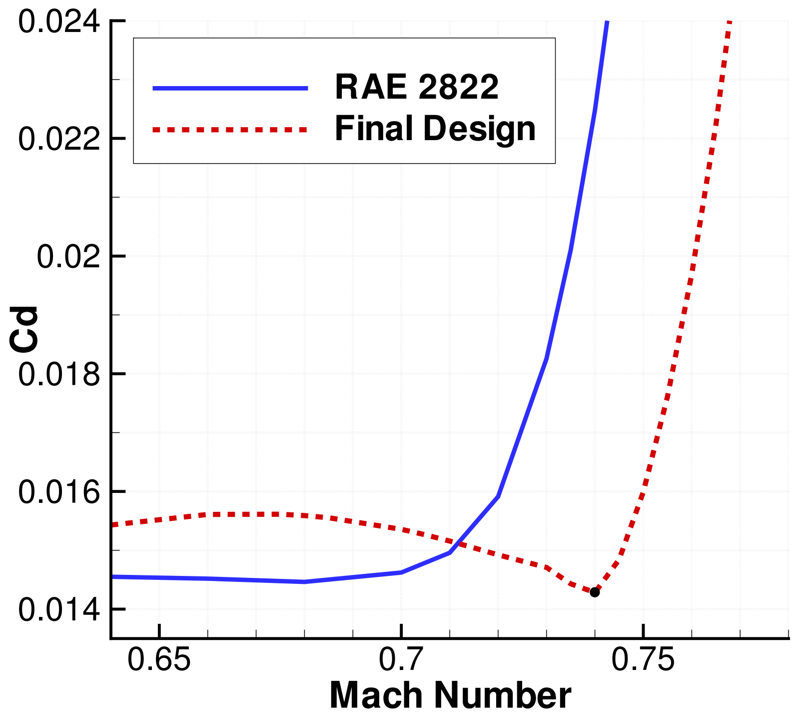
$$\mathcal{J} = \omega_{\mathrm{L}} \left(1 - \frac{C_{\mathrm{L}}}{C_{\mathrm{L}}^*}\right)^2 + \omega_{\mathrm{D}} \left(1 - \frac{C_{\mathrm{D}}}{C_{\mathrm{D}}^*}\right)^2 + \mathrm{T.C.}$$

- Design Point: $M_\infty = 0.74$, $\mathrm{Re} = 2.7 \times 10^6$
- $\omega_{\mathrm{L}} = 1.0$, $\omega_{\mathrm{D}} = 0.1$, $\omega_{\mathrm{T}} = 1.0$, T. Cons. @ 0.35, 0.96, 0.99 %c
- Targets: $C_{\mathrm{L}}^* = 0.733$, $C_{\mathrm{D}}^* = 0.013$
- 19 Geometric Design Variables + $\alpha$

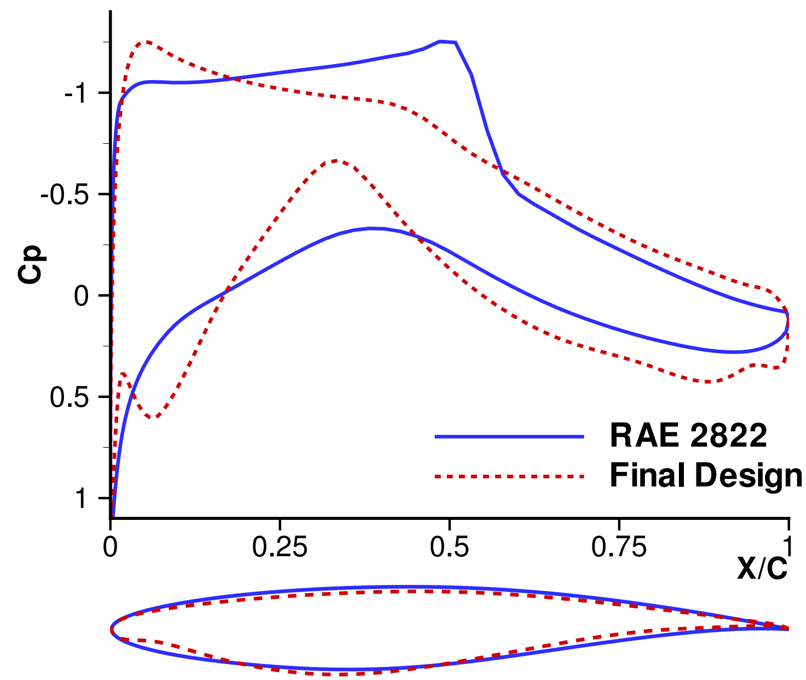


# Single-Point Lift-Constrained Drag Minimization at Transonic Speed

$$\mathcal{J} = \omega_{\mathrm{L}} \left(1 - \frac{C_{\mathrm{L}}}{C_{\mathrm{L}}^{*}}\right)^2 + \omega_{\mathrm{D}} \left(1 - \frac{C_{\mathrm{D}}}{C_{\mathrm{D}}^{*}}\right)^2 + \mathrm{T.C.}$$




Drag Reduction 36.4 %

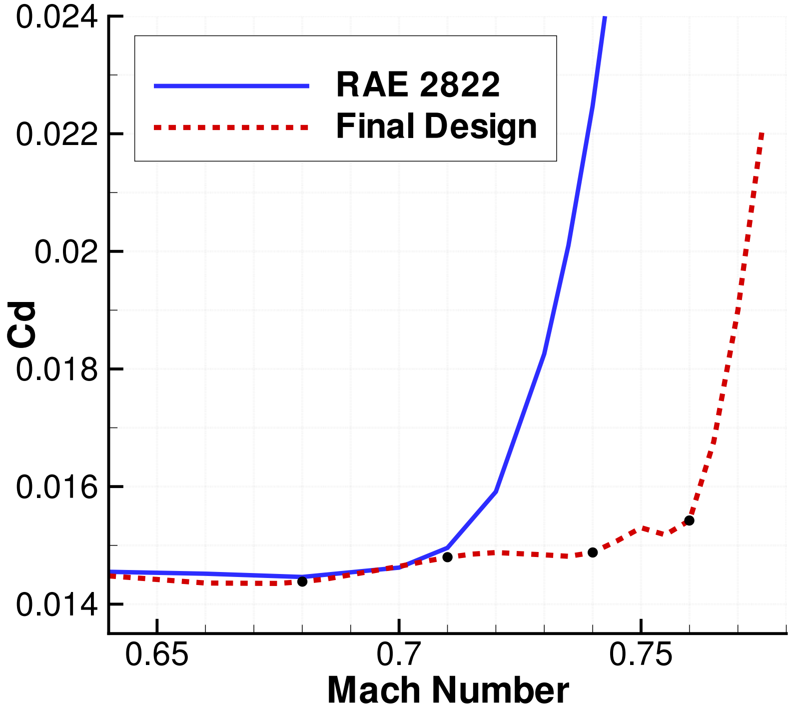


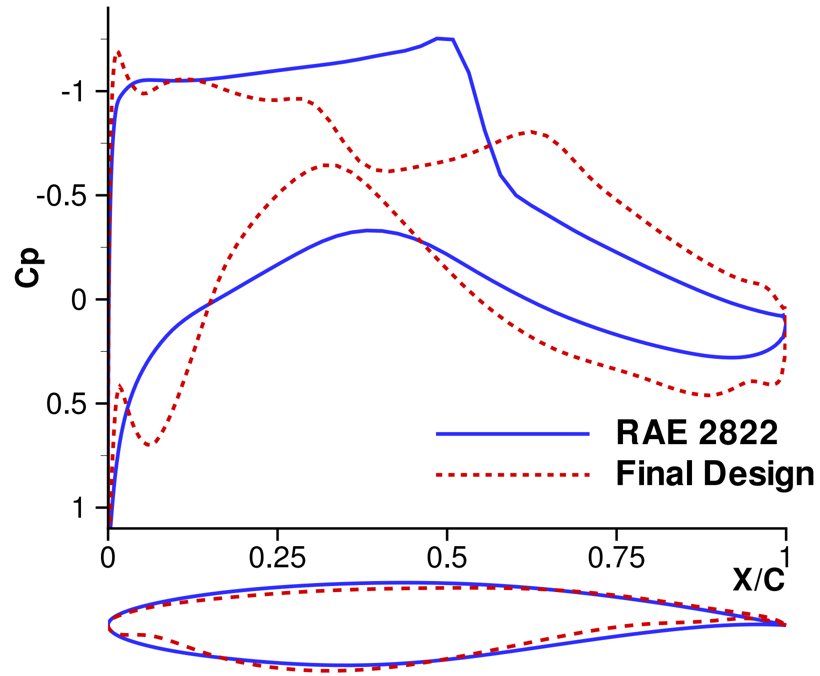

$M = 0.74$

Final Design

# Four-Point Lift-Constrained Drag Minimization at Transonic Speed

$$\mathcal{J}_{\mathrm{MP}} = \frac{1}{7}\mathcal{J}\left(M = 0.68\right) + \frac{1}{7}\mathcal{J}\left(M = 0.71\right) + \frac{2}{7}\mathcal{J}\left(M = 0.74\right) + \frac{3}{7}\mathcal{J}\left(M = 0.76\right)$$

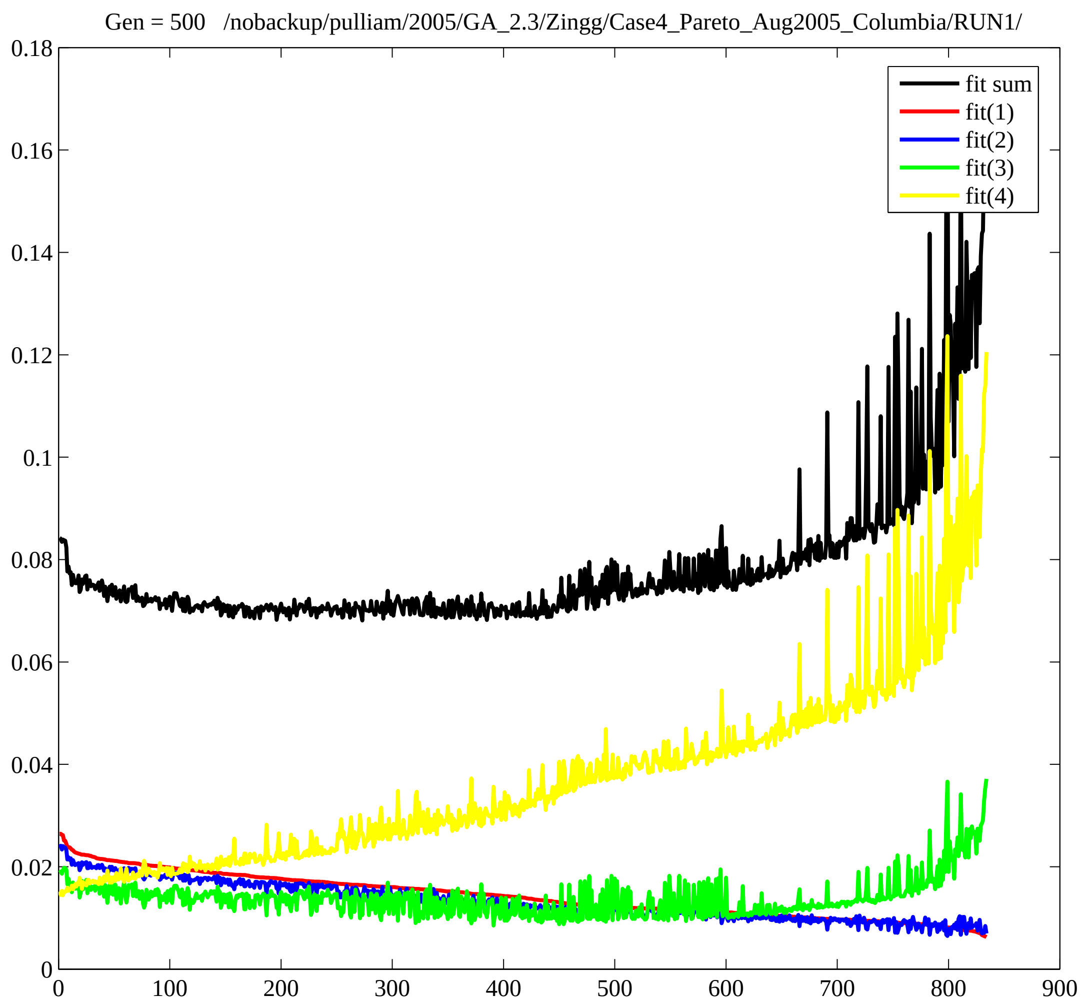


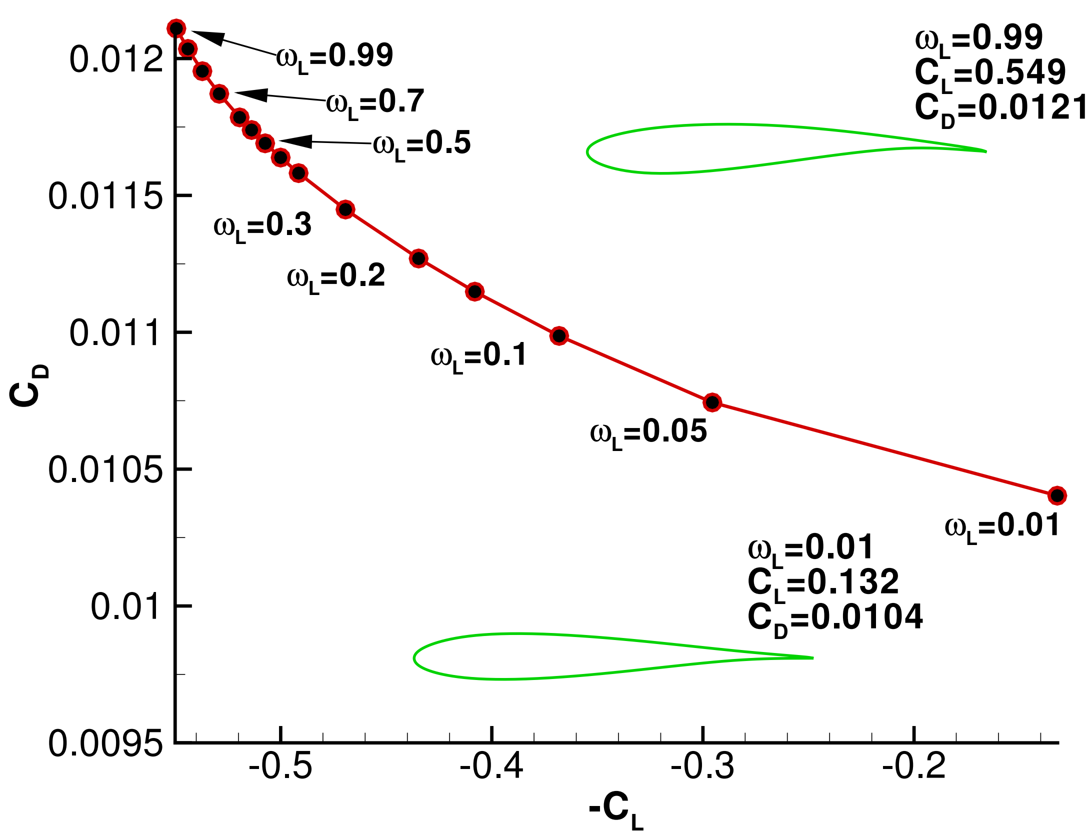

Gradient-based cost: 104 iterations, 832 equivalent function evaluations

GA Cost: $> 300$ generations, $> 23,000$ function evaluations

# Four-dimensional "Pareto Front"



# Multi-Objective Problems




**Pareto Front**

- Useful for studying trade-offs

- Competition among objectives: $\Rightarrow$ there is no unique optimum

- We seek a set of non-inferior solutions: $\Rightarrow$ Pareto front

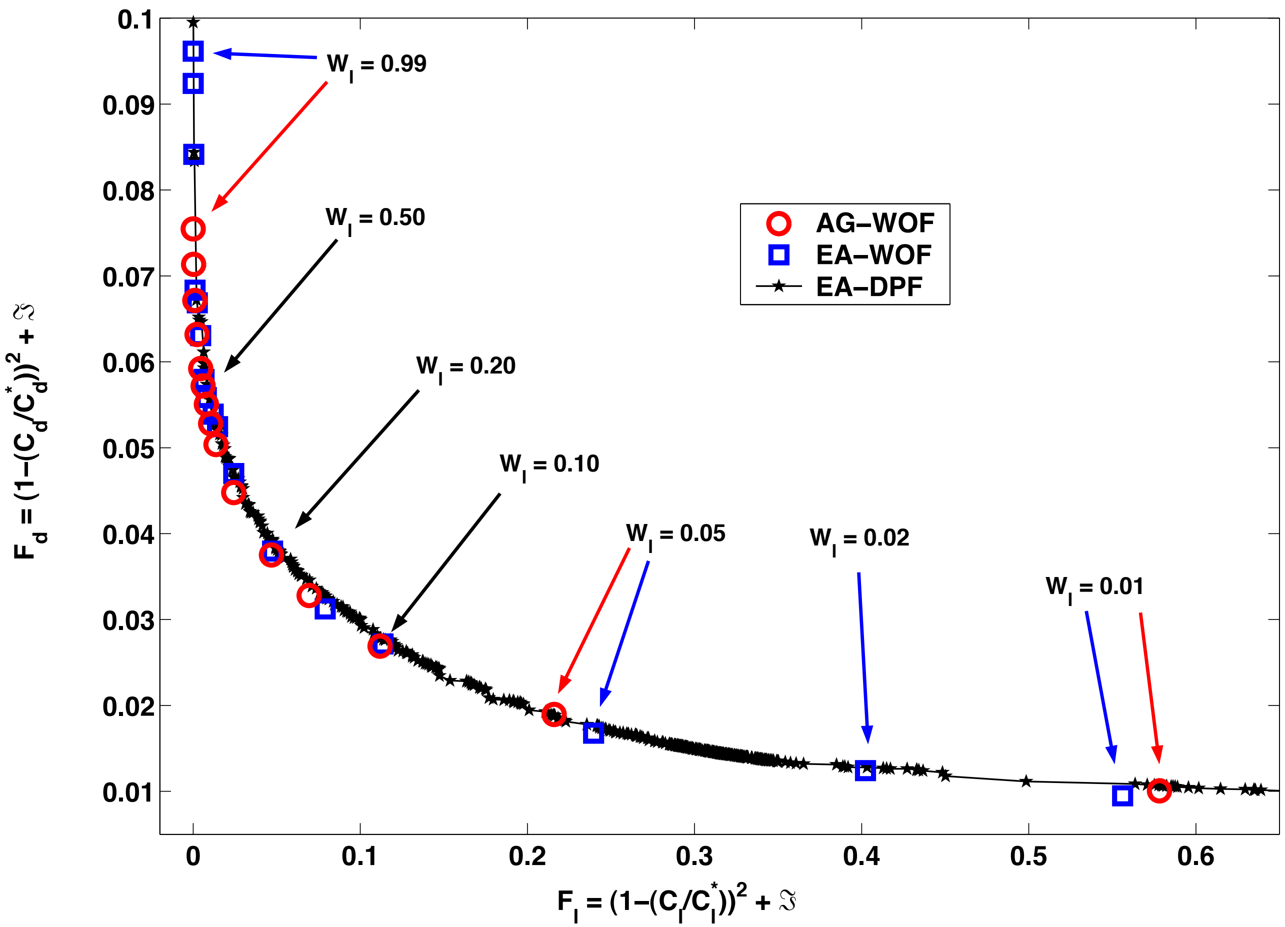
- Define two objectives:

  1. $f_1 = \left(1 - \frac{C_{\mathrm{D}}}{C_{\mathrm{D}}^*}\right)^2$
  2. $f_2 = \left(1 - \frac{C_{\mathrm{L}}}{C_{\mathrm{L}}^*}\right)^2$

- *Weighted Sum Method*:

$$\mathcal{J} = (1 - \omega_{\mathrm{L}})\, f_1 + \omega_{\mathrm{L}} f_2$$

# Comparison of Three Pareto Fronts




## Algorithm costs

Gradient-based: 2,438; Genetic (WOF): 70,000 equivalent function evaluations

# Convergence of DPF Approach

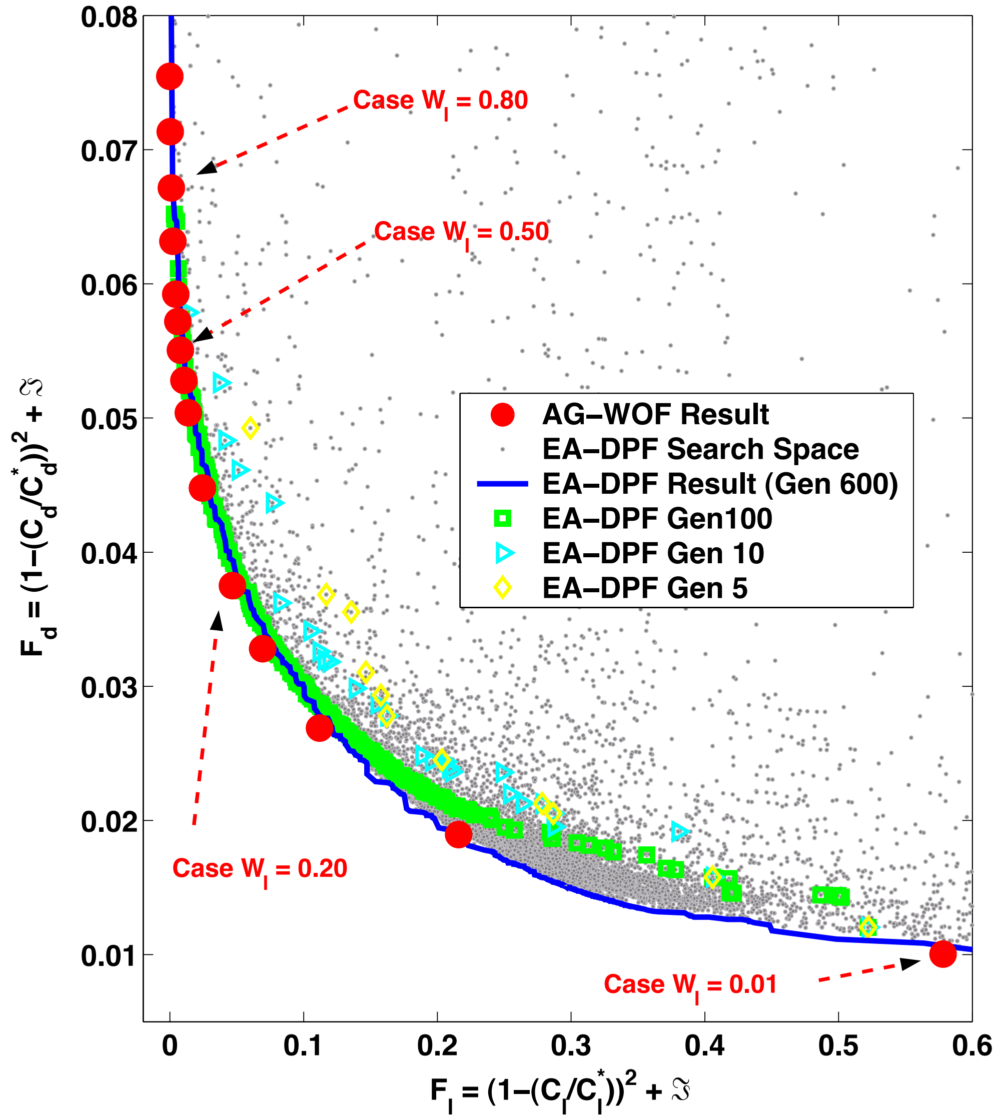
## DPF Costs

5 generations: 105 FE's

10 generations: 205 FE's

100 generations: 1999 FE's

600 generations: 12,000 FE's



# Topic I: Conclusions

- gradient-based algorithm is from $14$ to $> 100$ times faster than the GA depending on the number of design variables and the degree of convergence
  - gradient-based algorithm scales roughly linearly with the number of design variables
  - the GA's cost increases more rapidly as the number of design variables is increased
  - the relative cost of the GA increases substantially with tighter convergence requirements

- the GA is more suited to preliminary design (low-fidelity models, low convergence tolerance, trade-offs important)

- the gradient-based algorithm is more appropriate for detailed design (high-fidelity simulations, tight convergence tolerance, heavily constrained)

# Topic II:
# Optimization Under Variable Operating Conditions

- issues in problem formulation
  - can the optimization problem be posed *a priori*?
- an automated approach to the selection of weights and operating points

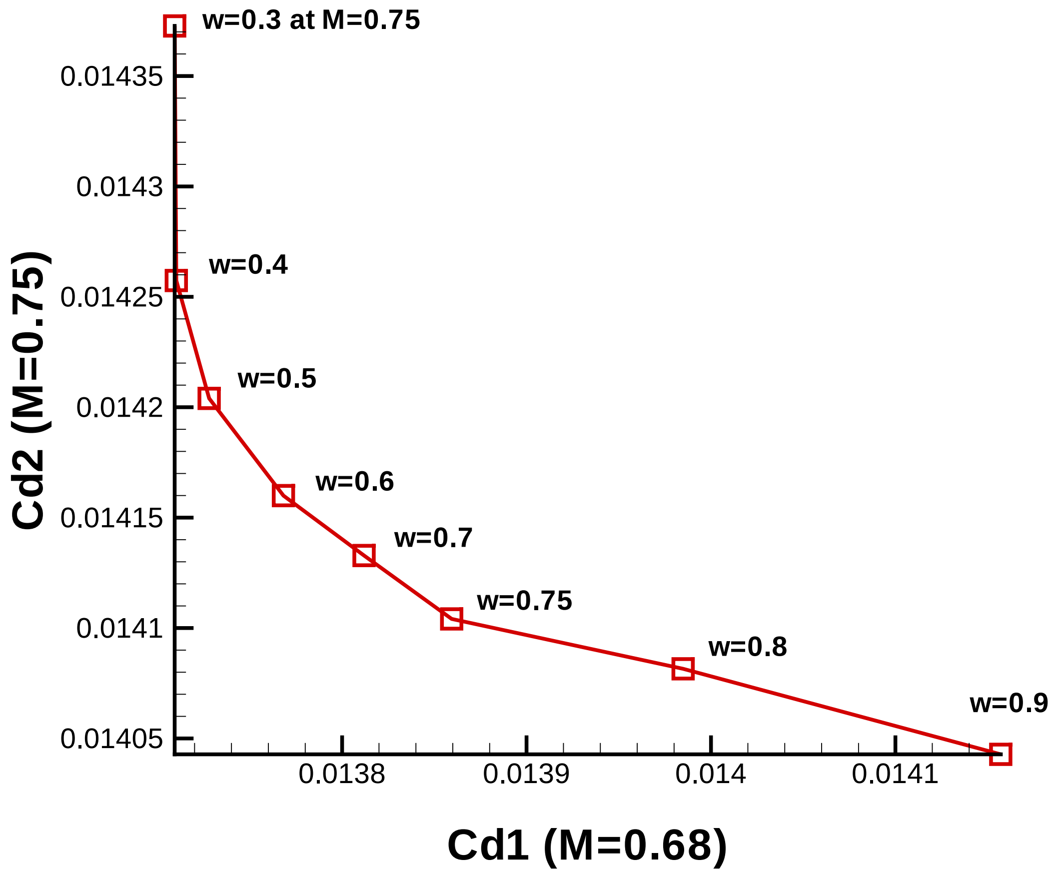
# Issues in Problem Formulation

- What is the objective?
  - minmax?
  - equal performance over the range of operating conditions?
  - minimize weighted integral?

- Role of off-design requirements
  - can dominate on-design performance
  - can be selected quite arbitrarily

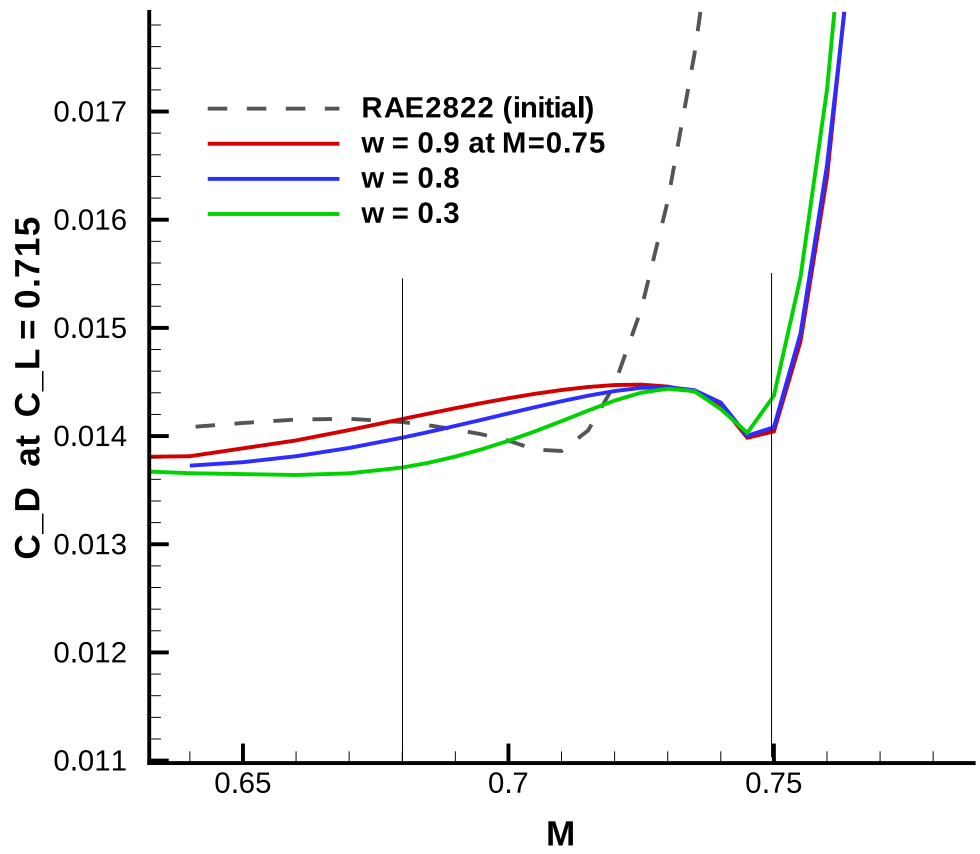
- Helpful to view the multi-point problem as a multi-objective problem

# Two-Point Lift-Constrained Drag Minimization



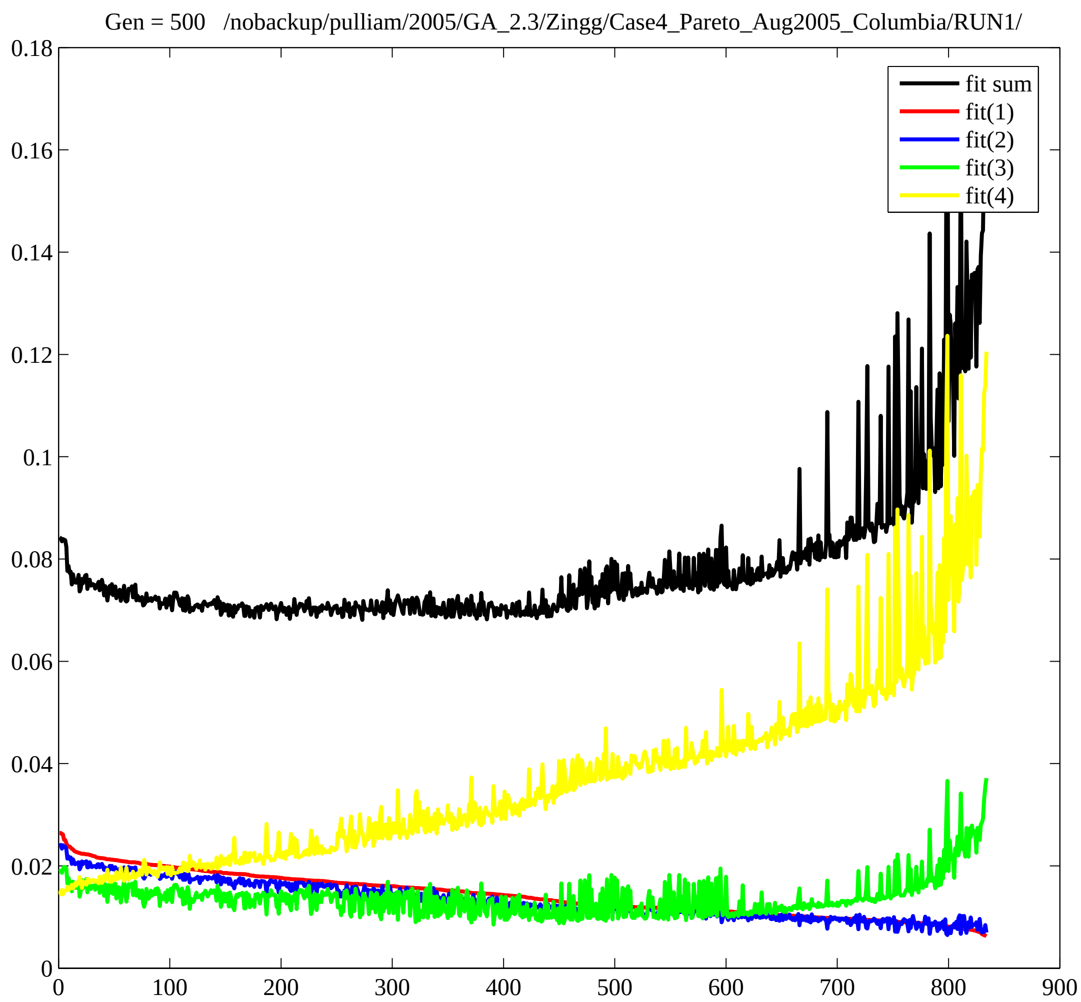

- Lift-constrained drag minimization, $C_l = 0.715$
- Two operating points: $M = 0.68$, $M = 0.75$

# Two-Point Lift-Constrained Drag Minimization




- Lift-constrained drag minimization, $C_l = 0.715$
- Two operating points: $M = 0.68$, $M = 0.75$

# Four-dimensional “Pareto Front”



# Automated Approach

- automated weight specification

$$w_i^{new} = w_i^{old} + c \left( \frac{C_{Di}}{\sum_{i=1}^{N} C_{Di}} - \frac{1}{N} \right)$$

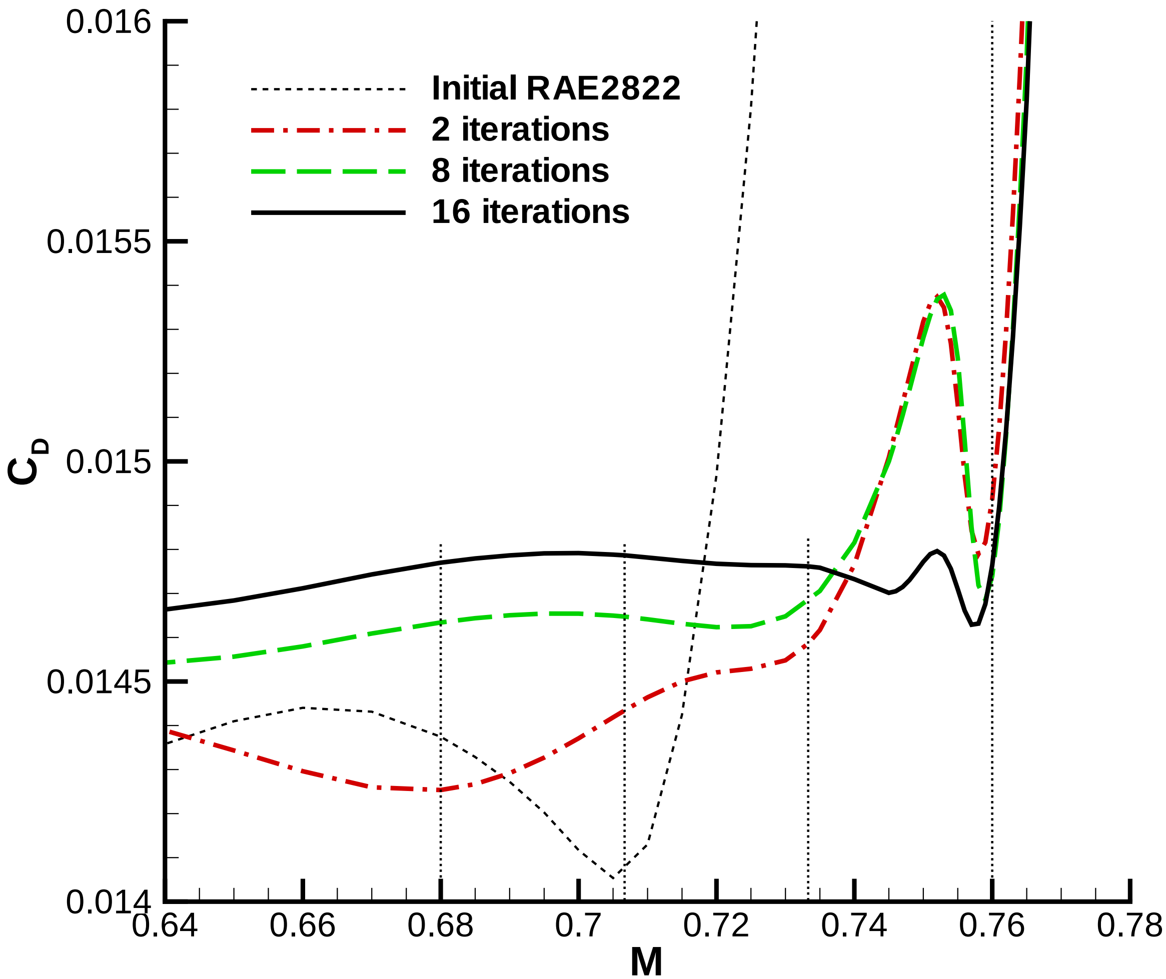
  - weights must be non-negative
  - operating points can be dropped if the weight is zero

- automated selection of operating points
  - periodically examine performance between operating points
  - add new operating points at significant local maxima if they exist

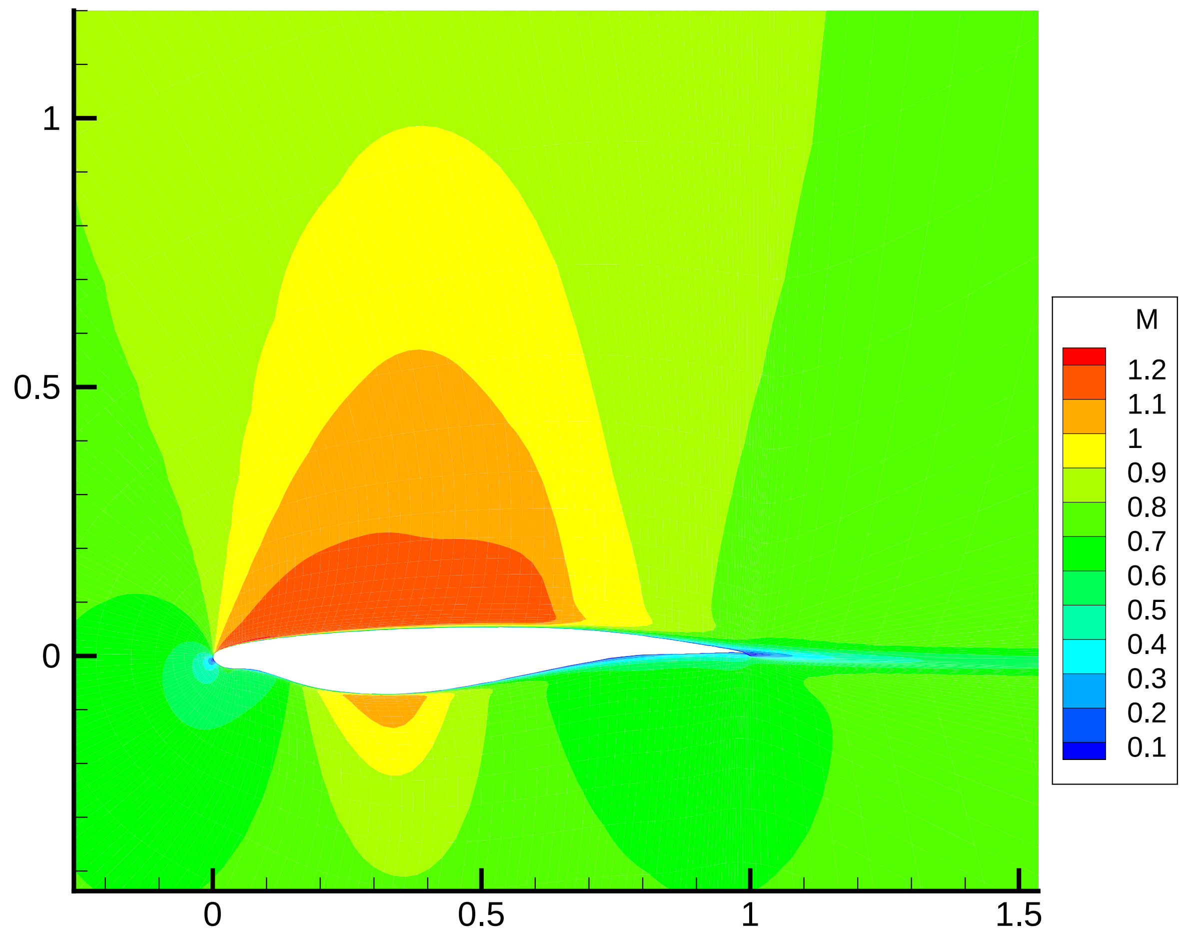
# Lift-Constrained Drag Minimization at Transonic Speed

- lift-constraint: $C_l = 0.715$
- Mach number range from 0.68 to 0.76
- Reynolds number: 9 million
- initial airfoil: RAE 2822
- 23 design variables, including angle of incidence
- 4 thickness constraints
- initial operating points: $M = 0.68,\ 0.70667,\ 0.73333,\ 0.76$
- initial weights: 0.25, 0.25, 0.25, 0.25

# Lift-Constrained Drag Minimization over a Mach Number Range



# Mach Number Contours at $M = 0.76, C_l = 0.733$



# Drag Coefficients at Operating Points

| | $C_D$ | ($C_L$=0.733) | | | | |
|---|---|---|---|---|---|---|
| | **0.68** | **0.70667** | **0.7333** | **0.752** | **0.76** | **St Dev** |
| 1 | 0.014176 | 0.014391 | 0.014521 | | 0.015211 | 0.0004473 |
| 2 | 0.014254 | 0.014435 | 0.014586 | | 0.014915 | 0.0002801 |
| 3 | 0.014351 | 0.014484 | 0.014576 | | 0.014873 | 0.0002216 |
| 4 | 0.014683 | 0.014652 | 0.014611 | | 0.014769 | 0.0000668 |
| 5 | 0.014720 | 0.014679 | 0.014632 | | 0.014757 | 0.0000538 |
| 6 | 0.014779 | 0.014698 | 0.014649 | | 0.014736 | 0.0000553 |
| 7 | 0.014750 | 0.014682 | 0.014642 | | 0.014738 | 0.0000503 |
| 8 | 0.014634 | 0.014647 | 0.014681 | 0.015378 | 0.014740 | 0.0003169 |
| 9 | 0.015039 | 0.014897 | 0.014694 | 0.014918 | 0.014717 | 0.0001452 |
| 10 | 0.014745 | 0.014768 | 0.014718 | 0.014901 | 0.014756 | 0.0000713 |
| 11 | 0.014784 | 0.014800 | 0.014729 | 0.014847 | 0.014754 | 0.0000452 |
| 12 | 0.014781 | 0.014804 | 0.014751 | 0.014818 | 0.014755 | 0.0000294 |
| 13 | 0.014784 | 0.014798 | 0.014749 | 0.014806 | 0.014759 | 0.0000246 |
| 14 | 0.014782 | 0.014794 | 0.014757 | 0.014795 | 0.014760 | 0.0000182 |
| 15 | 0.014777 | 0.014787 | 0.014760 | 0.014794 | 0.014763 | 0.0000150 |
| **16** | **0.014770** | **0.014787** | **0.014762** | **0.014786** | **0.014764** | **0.0000119** |

# Evolution of Weights

| Iteration | c | Weights 0.68 | (Mach No.) 0.70667 | 0.73333 | 0.753 | 0.76 |
|---|---|---|---|---|---|---|
| 1 | | 0.25000 | 0.25000 | 0.25000 | | 0.25000 |
| 2 | 15 | 0.14736 | 0.20276 | 0.23624 | | 0.41365 |
| 3 | 15 | 0.07167 | 0.17371 | 0.24624 | | 0.50837 |
| 4 | 15 | 0.01497 | 0.15141 | 0.24749 | | 0.58613 |
| 5 | 15 | 0.01598 | 0.14456 | 0.23031 | | 0.60915 |
| 6 | 15 | 0.02188 | 0.13996 | 0.21372 | | 0.62444 |
| 7 | 15 | 0.03808 | 0.13543 | 0.19683 | | 0.62966 |
| 8 | 15 | 0.04999 | 0.13010 | 0.18127 | 0.00000 | 0.63865 |
| 9 | 15 | 0.01311 | 0.09592 | 0.15396 | 0.11384 | 0.62317 |
| 10 | 15 | 0.05062 | 0.10483 | 0.12189 | 0.12695 | 0.59571 |
| 11 | 15 | 0.04409 | 0.10294 | 0.10973 | 0.15193 | 0.59131 |
| 12 | 15 | 0.04441 | 0.10647 | 0.09879 | 0.16495 | 0.58539 |
| 13 | 15 | 0.04430 | 0.11093 | 0.09259 | 0.17228 | 0.57990 |
| 14 | 15 | 0.04529 | 0.11475 | 0.08648 | 0.17772 | 0.57576 |
| 15 | 15 | 0.04622 | 0.11815 | 0.08234 | 0.18116 | 0.57214 |
| **16** | **15** | **0.04637** | **0.12041** | **0.07901** | **0.18481** | **0.56940** |

# Optimization Under Variable Operating Conditions
# Conclusions

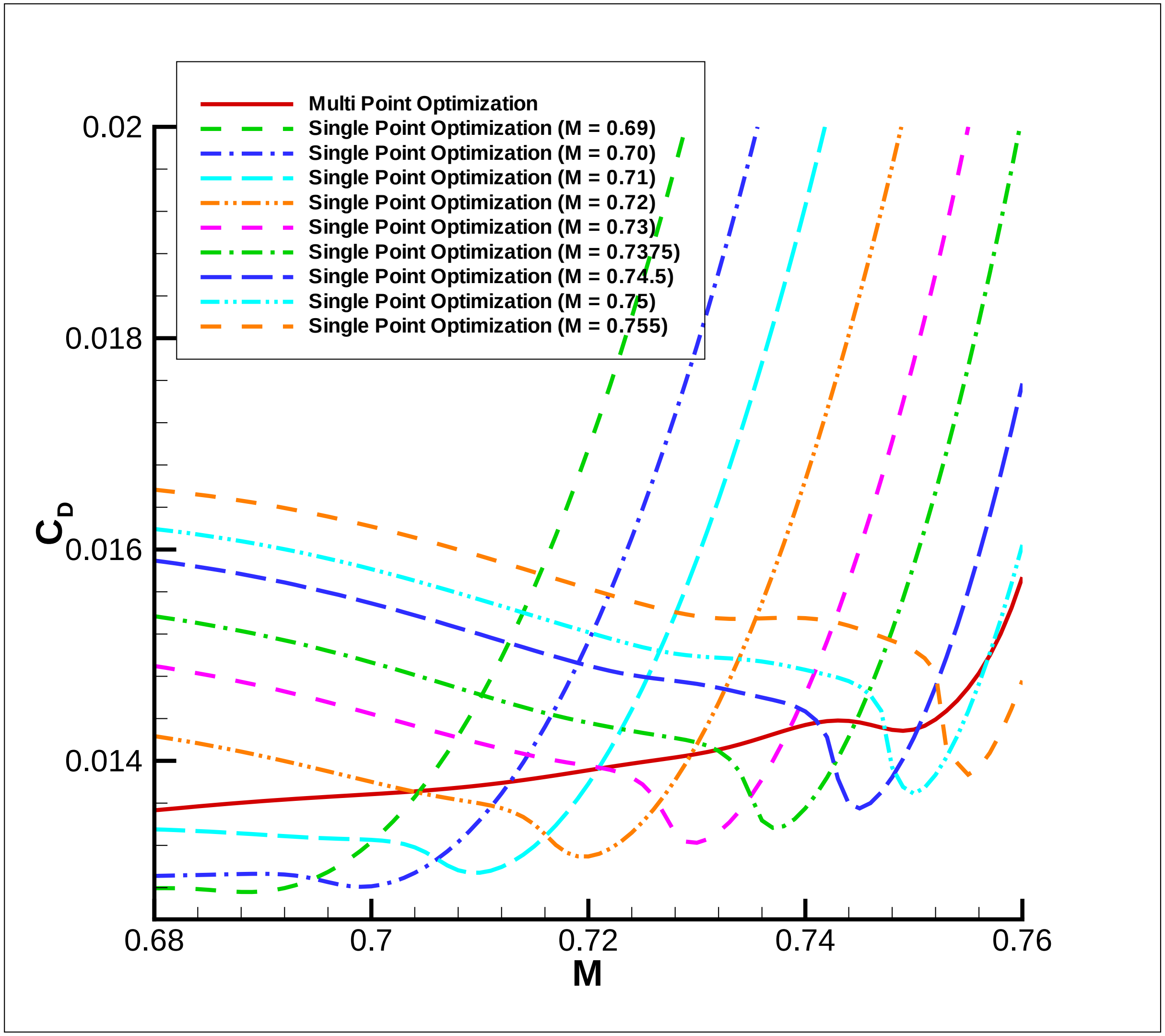
- Can the optimization problem be posed *a priori*? $\Rightarrow$ No, some knowledge of trade-offs is essential for proper problem specification

- properly chosen weights coupled with automatic introduction of additional operating points at local maxima will minimize a weighted integral

- if this leads to inadequate performance at some points, then the weights must be modified

- desirable aerodynamic performance must be specified much more precisely than previously

# Or ... Morphing Airfoils?



# Or ... Morphing Airfoils?