Model Management in Design and MDO Problem Synthesis & Solution

Natalia Alexandrov

ASCAC Methods Development Peer Review

28-29 November 2001

n.alexandrov@larc.nasa.gov

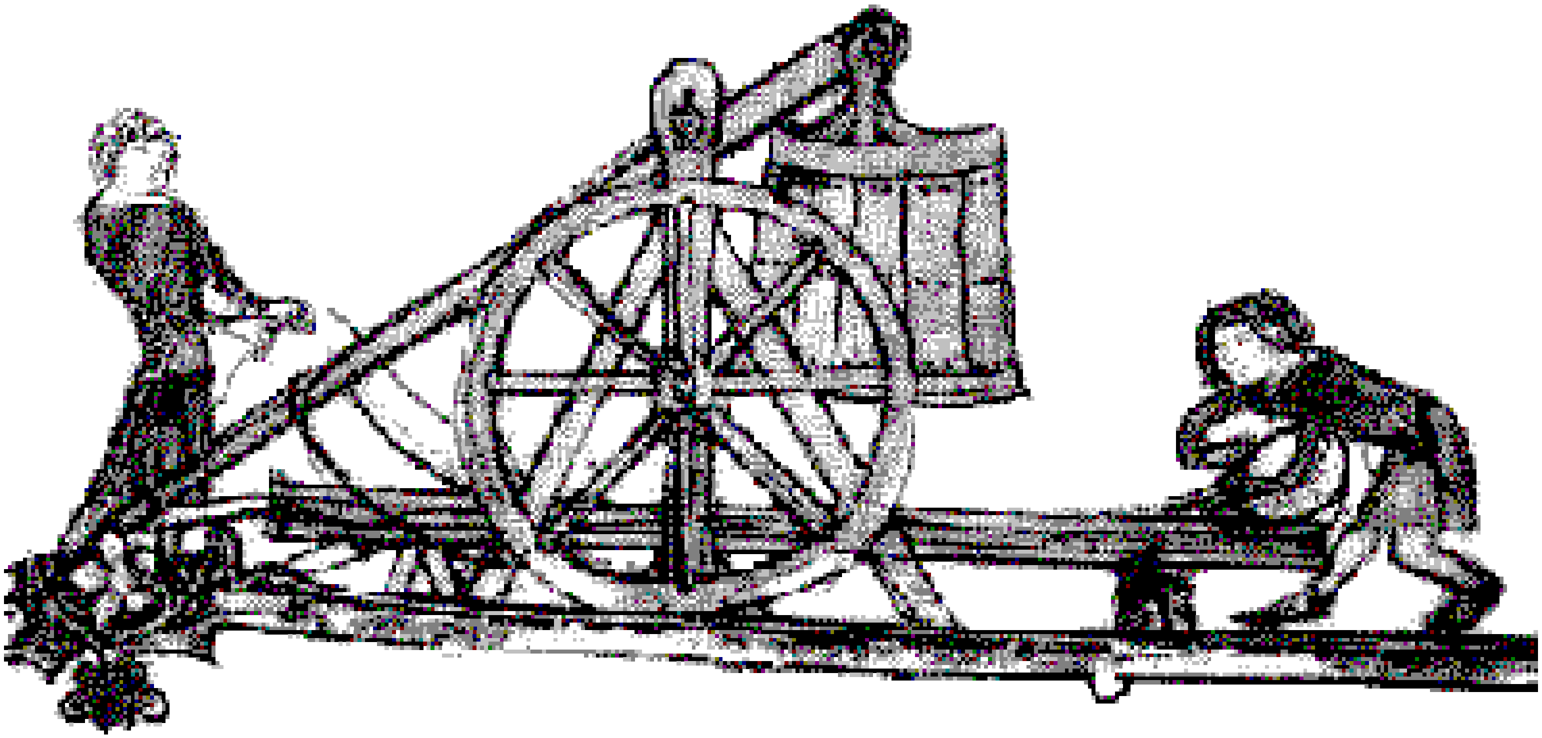
Some Major Technical Obstacles to MDO

- **Modeling**
  - Analysis-based functions are expensive and not computationally robust
  - Difficult to obtain reliable and affordable derivatives
- **Optimization**
  - Desired disciplinary autonomy is in conflict with robust and efficient optimization
  - Algorithms for analysis-based design are in their infancy

Outline

- **Part I: Approximation and Model Management Optimization (AMMO) Framework**
  - **Optimization strategy**
    - **Reduce cost of using high-fidelity analyses**
    - **Maintain convergence to high-fidelity answers**
- **Part II: MDO Problem Synthesis and Solution**
  - **MDO problem formulations and attendant optimization algorithms**
    - **Preserve maximum disciplinary autonomy**
    - **Solve the problem reliably and efficiently**

Part I:
Approximation and Model Management Optimization (AMMO)



ASCoT Project (1998-2002)
(Aerospace Systems Concept to Test)

Project Vision

Physics-based modeling and simulation with sufficient speed and accuracy for validation and certification of advanced aerospace vehicle design in less than 1 year

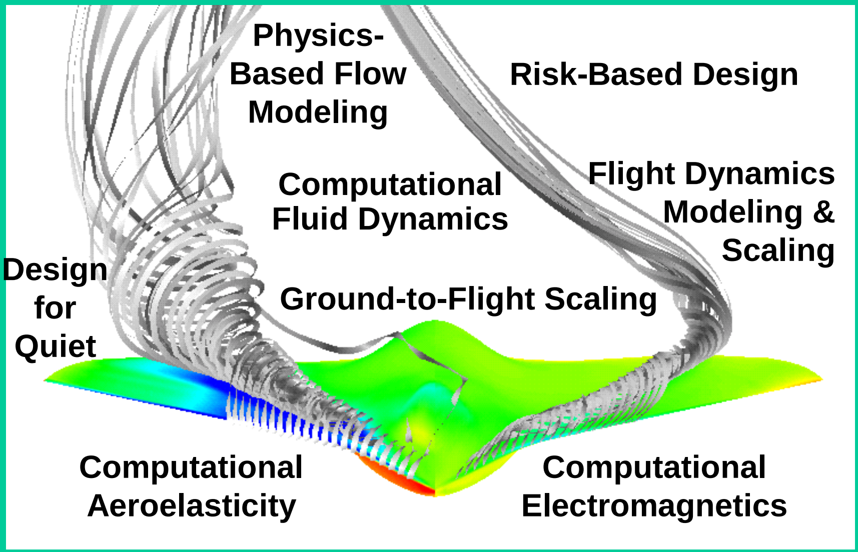
Project Goal

- **Provide next-generation analysis & design tools to increase confidence and reduce development time in aerospace vehicle designs**

Objective

- Develop **fast, accurate, and reliable** analysis and design tools via fundamental technological advances in:
  - Physics-Based Flow Modeling
  - **Fast, Adaptive, Aerospace Tools (FAAST)** (CFD and Design)
  - Ground-to-Flight Scaling
  - Time-Dependent Methods
  - Design for Quiet
  - Risk-Based Design

Technology Areas



Benefit

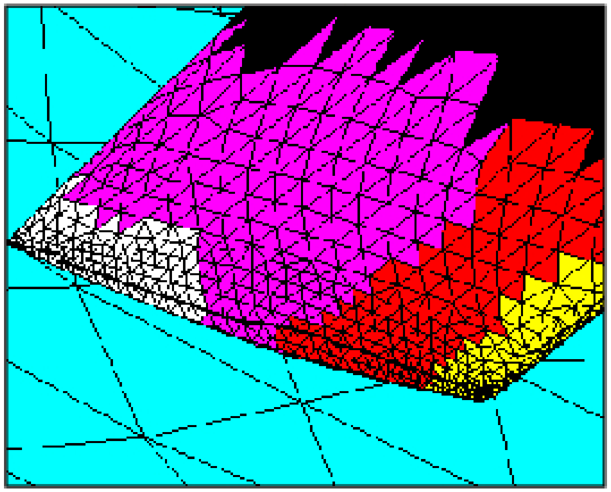
- Increased Design Confidence
- Reduced Development Time

Collaborators
(chronological order)

FY96-present	N.M. Alexandrov (MDOB)	Algorithms and demonstrations
FY97-00	R.M. Lewis (ICASE)	Algorithms
FY98-99	C.R. Gumbert (MDOB)	Variable-resolution CFL3D models
FY99	L.L. Green (MDOB)	Variable-resolution FLOMG models
FY98-99	P.A. Newman (MDOB)	CFD consulting
FY00	W.K. Anderson (AAAC/CSMB)	FUN2D, FUN3D, adjoint solvers
FY01–present	E.J. Nielsen (AAAC/CSMB)	Variable-physics FUN2D/FUN3D models
FY01–present	M.A. Park (AAAC/CSMB)	Mesh adaptation
FY01–present	A. Yates (W & M student)	Derivative-free methods
FY02–present	J.A. Samareh (MDOB)	Geometry parameterization

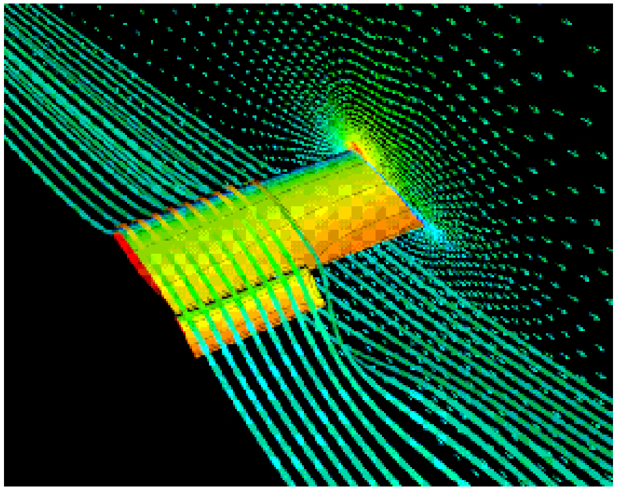
Current Design Environment

Domain Decomposition
(parallel processing)



Adjoint
Solver

Flow Solvers
(FUN2D, FUN3D)

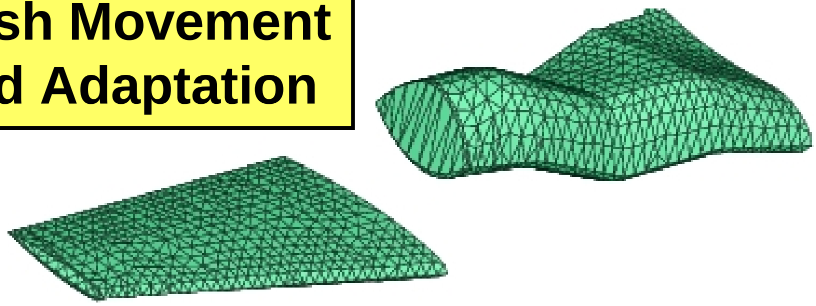


Optimization

Derivative
Evaluation

Parameterization
(MASSOUD)

Mesh Movement
and Adaptation



Limiting factors

- **Extreme expense of repeated analyses**
  - **Example: turbulent computation on 1 M grid points (Nielsen and Anderson)**
    - **1 day for submission, 3-4 days in queue**
    - **8 hours per 1 design cycle on 112 CPU**
      - **Flow solution**
      - **Adjoint solution with 20-39 grid sensitivities and gradient evaluations**
      - **Line search with 5-6 grid moves and flow solutions**
    - **10 design cycles ≈ 9000 CPU hours for a simple single-point design**
- **Function and derivative evaluations prone to failure away from the nominal design**
- **Derivative-free optimization is not an option due to computational expense**

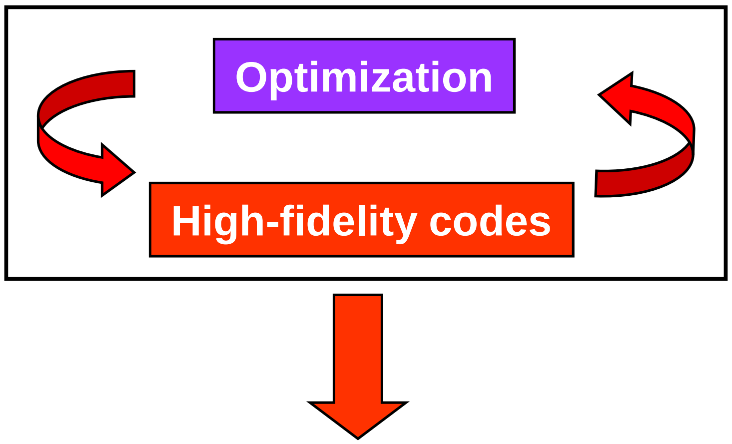
Approach

- **Engineering**
  - **A variety of approximations and models available and used for a long time**
  - **Ad hoc optimization techniques**
- **Mathematical programming**
  - **Generally limited to local Taylor series models**
  - **Rigorous and robust optimization techniques**
- **AMMO**
  - **Use of engineering approximations and models**
  - **Rigorous and robust optimization techniques**
  - **Can be used with any gradient-based algorithm**
- **Modeling and grid difficulties also being addressed**

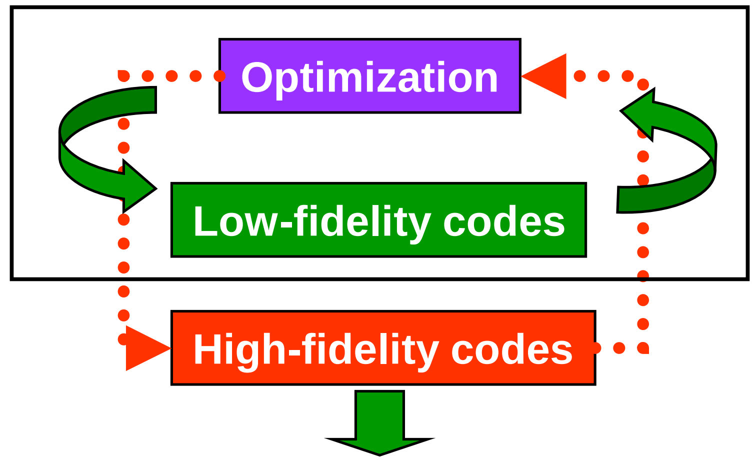
AMMO Idea

Objective: reduce cost of design optimization with analyses

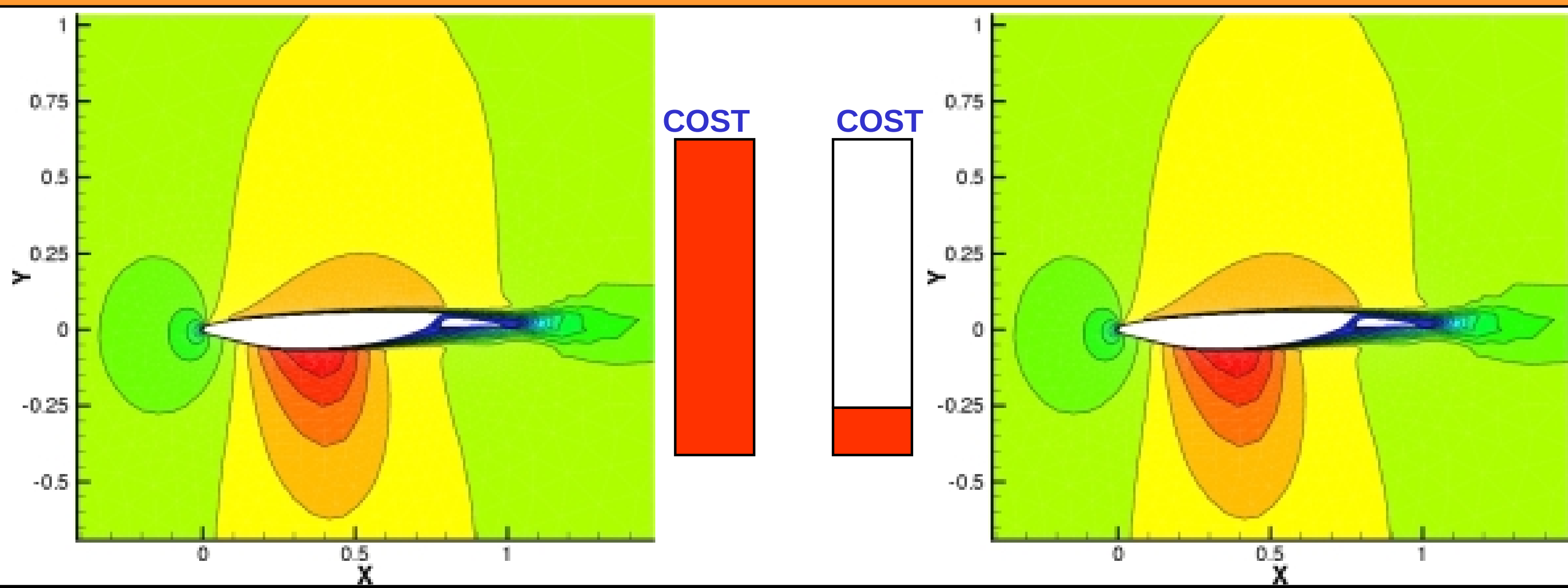
Conventional Optimization



AMMO



Multi-Element Airfoil Design: AMMO Demonstration for Navier-Stokes / Euler CFD



AMMO gives Navier-Stokes answers with five-fold savings

Design Optimization Problem

- **The analysis problem:** **Given x, solve system**

$$A(x,u(x)) = 0$$

for u that describes the physical behavior of the system

- **The design problem** **(canonical formulation): Solve**

$$\begin{aligned} \text{minimize} \quad & f(x, u(x)) \\ \text{subject to} \quad & c_i(x, u(x)) = 0,\ i \in E \\ & c_i(x,u(x)) \leq 0,\ i \in I \\ & x_L \leq x \leq x_U \end{aligned}$$

where, given x, u(x) is determined from $A(x,u(x)) = 0$

Convergence vs. Performance

- **Convergence relies on ensuring local similarity of trends**
  - Let $\tilde{f}$ be some lower-fidelity model of f. At each major iteration k, $\tilde{f}$ is required to satisfy

$$\tilde{f}(x_k) = f(x_k), \quad \nabla \tilde{f}(x_k) = \nabla f(x_k)$$

    Easily enforced when derivatives are available.

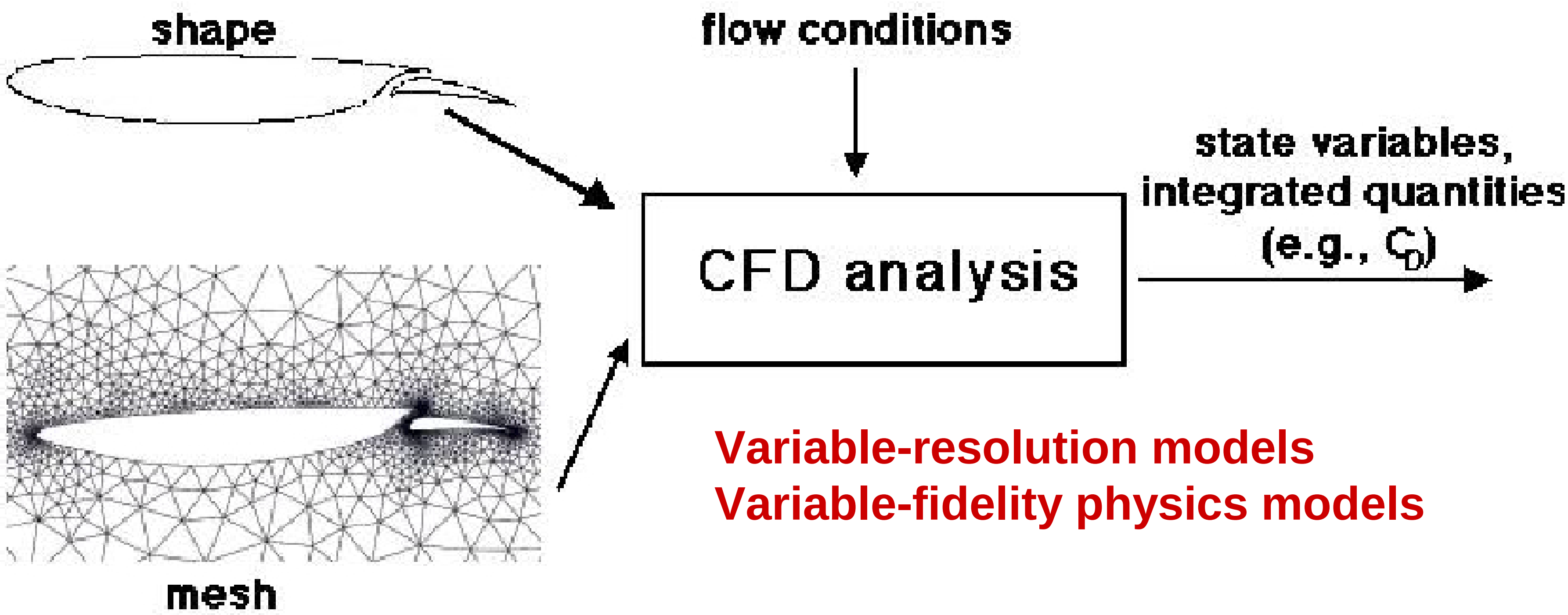
- **Enforcing first-order consistency: multiplicative β-correction, Chang et al. 1993**
  - Given $f(x)$ and $f_{lo}(x)$, define $\beta(x) \equiv \frac{f(x)}{f_{lo}(x)}$
  - Given x_k, build $\beta_k(x) = \beta(x_k) + \nabla \beta(x_k)^T (x - x_k)$
  - Then $\tilde{f}_k(x) = \beta_k(x) f_{lo}(x)$ satisfies the consistency conditions at x_k

- **Practical efficiency is problem/model dependent and is influenced by the ability to transfer computational load onto low-fidelity computation; at worst, AMMO is conventional optimization.**

Variable-Fidelity Models Used in AMMO

- **Variable accuracy**
  - Converge analyses to user-specified tolerance
- **Variable resolution**
  - Single physical model on meshes of varying degree of refinement
- **Variable-fidelity physics**
  - E.g., in CFD, physical models range from inviscid, irrotational, incompressible flow to Navier-Stokes equations for viscous flow
- **Other**
  - Data-fitting models, reduced-order models
- **Study favorable and unfavorable relationship between models**

Demonstration Problems: Aerodynamic Optimization



Variable-resolution models
Variable-fidelity physics models

minimize Integrated quantities, such as $-\frac{L}{D}$ $\left(\frac{\text{lift}}{\text{drag}}\right)$ or C_D (drag coefficient)

subject to constraints on, e.g., pitching and rolling moment coefficients, etc.

$x_l \leq x \leq x_u$

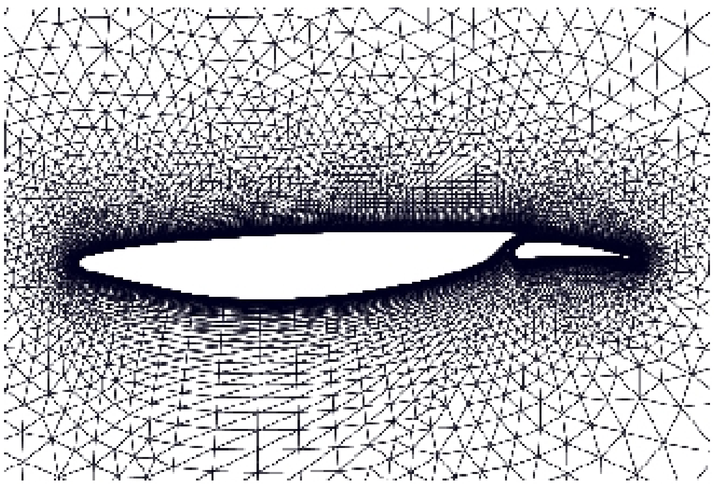
Managing Variable-Fidelity Physics Models: Multi-Element Airfoil
AIAA-2000-4886, Alexandrov, Nielsen, Lewis, Anderson

- **A two-element airfoil designed to operate in transonic regime – inclusion of viscous effects is important**
- **Governing equations – time-dependent Reynolds-averaged Navier-Stokes**
- **Flow solver – FUN2D, unstructured mesh methodology (Anderson, 1994)**
- **Sensitivity derivatives – discrete adjoint approach (Anderson, 1997)**
- **Conditions:**
  - $M_\infty = 0.75$
  - $Re = 9 \times 10^9$
  - $\alpha = 1^o$ **(global angle of attack)**

Multi-Element Airfoil, cont.

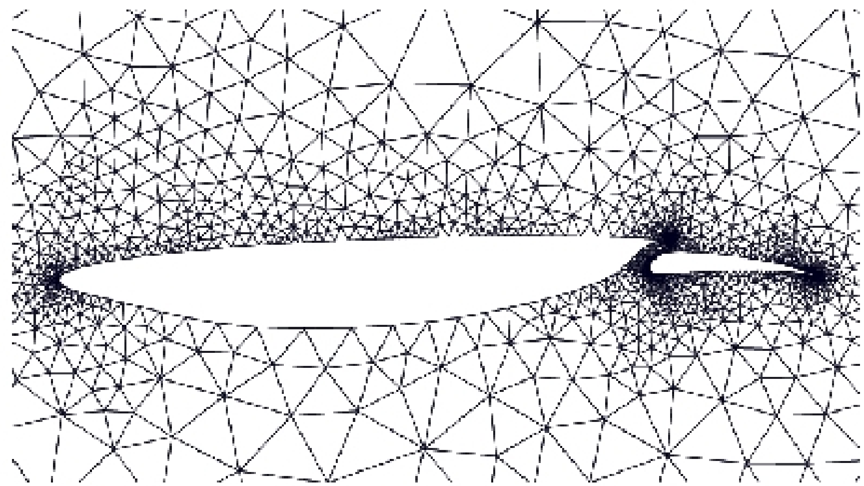
- Hi-fi model – FUN2D analysis in RANS mode
- Lo-fi model – FUN2D analysis in Euler mode
- Computing on SGI Origin^{TM} 2000, 4 R1OK processors

Viscous mesh:
10449 nodes and 20900 triangles



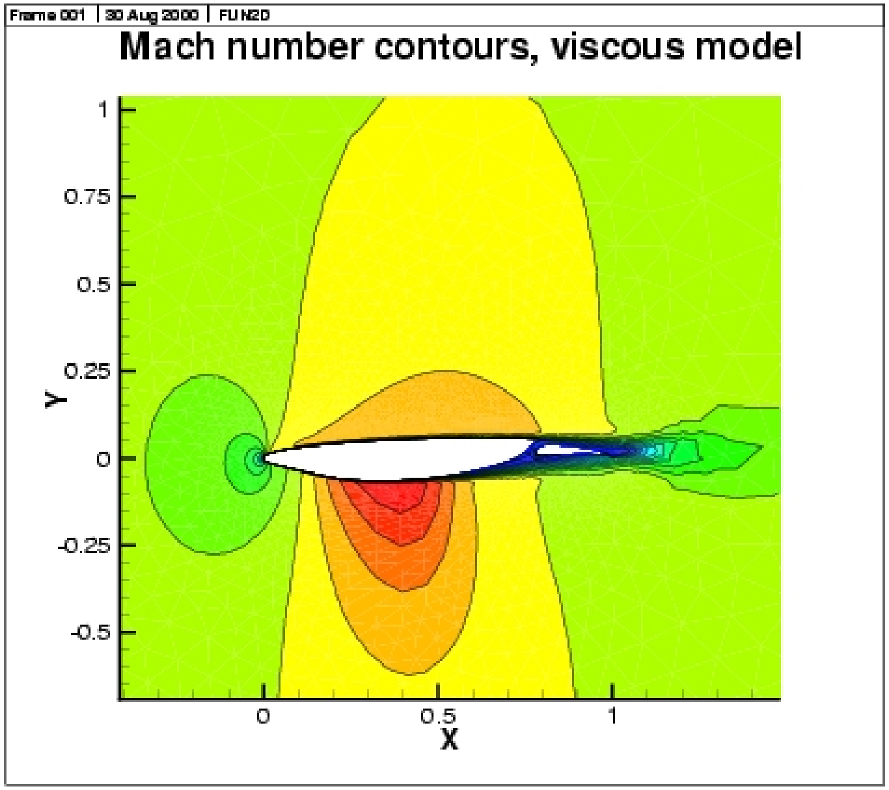
t/analysis $\approx$ 21 min
t/sensitivity $\approx$ 21 or 42 min

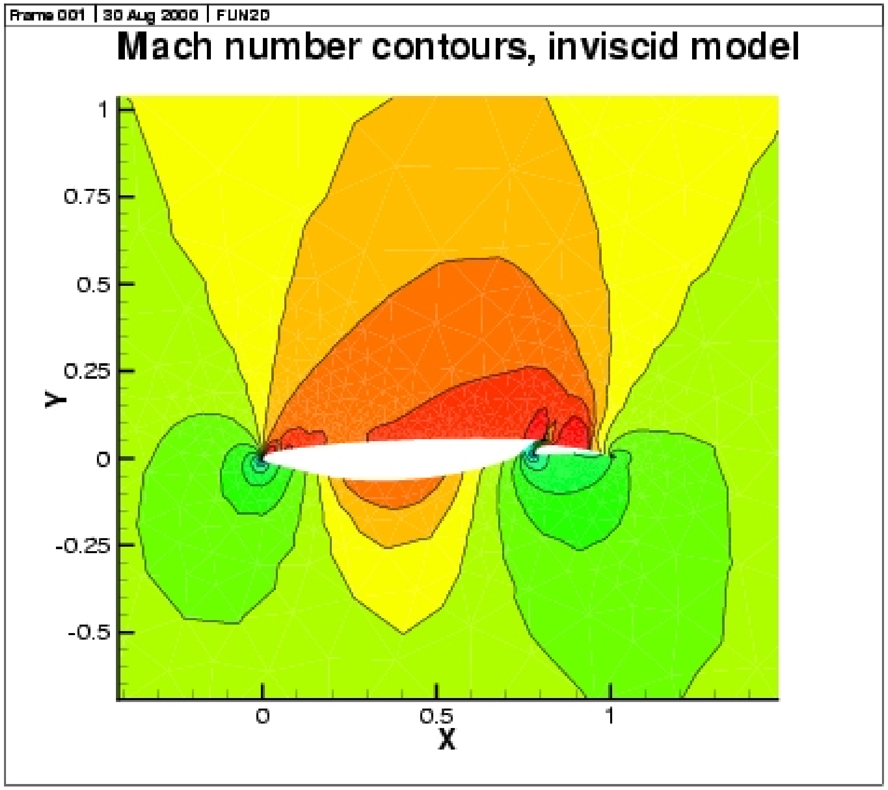
Inviscid mesh:
1947 nodes and 3896 triangles



t/analysis $\approx$ 23 sec
t/sensitivity $\approx$ 100 or 77 sec

Multi-Element Airfoil: Viscous Effects





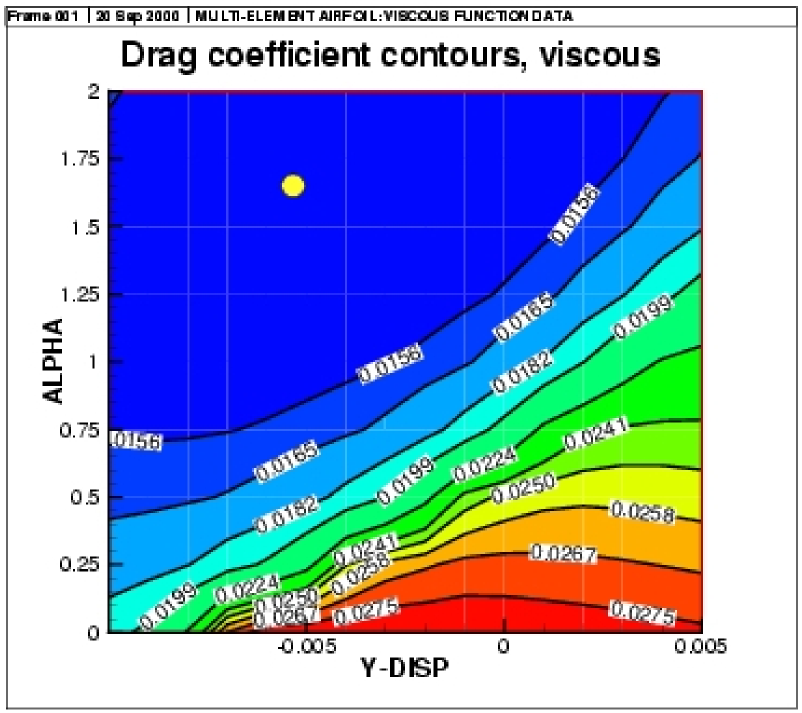
- **Boundary and shear layers are visible in the viscous case.**

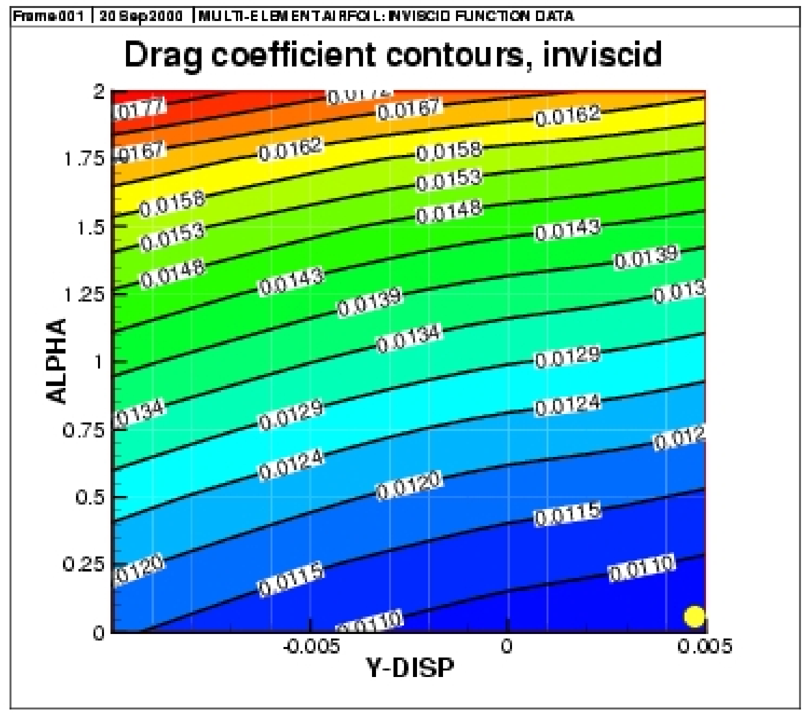
Multi-Element Airfoil: Computational Experiments

- **Objective function:** minimize drag coefficient subject to bounds on variables
- **Case 1:** (for visualization)
  - Variables: angle of attack, y-displacement of the flap
  - Solve problem with hi-fi models alone using a commercial optimization code (PORT, Bell Labs)
  - Solve the problem with AMMO, PORT used for lo-fi subproblems
- **Case 2:**
  - Variables: angle of attack, y-displacement of the flap, geometry description of the airfoil; 84 variables total
  - Same experiment

Multi-Element Airfoil: Models

- Time/function for inviscid model negligible compared to viscous model
- Descent trends are reversed — unusual but a good test

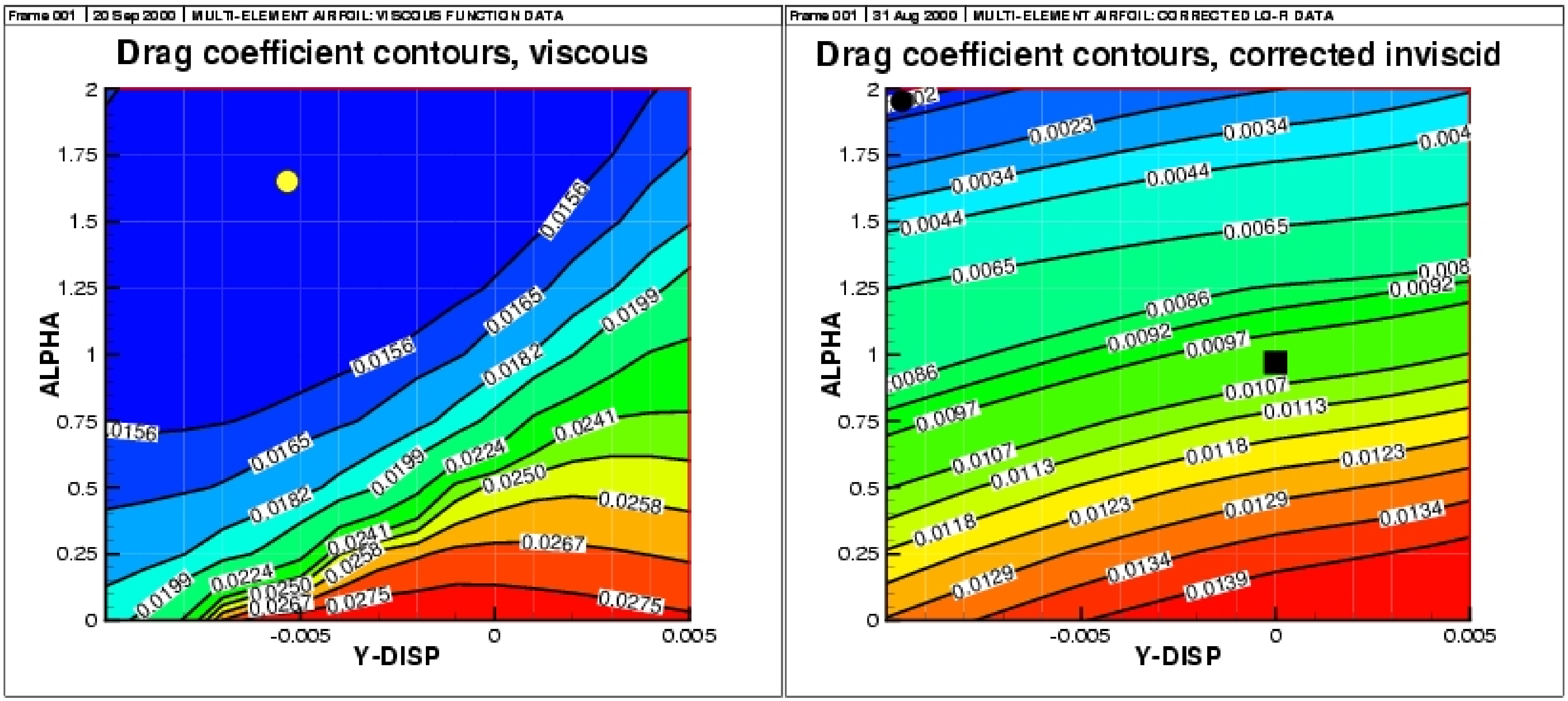




Multi-Element Airfoil: AMMO Iterations with 2 Variables

Iteration 1. **Starting point:** $\alpha = 1.0$, **y-disp** $= 0.0$

High-fidelity objective vs. corrected low-fidelity objective



New point: $\alpha = 2.0$, **y-disp** $= -0.01$

Multi-Element Airfoil: Performance Summary

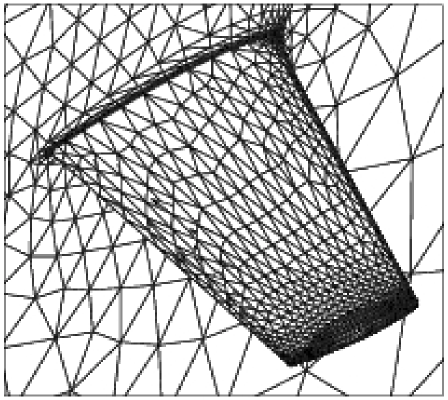
Notation: No. functions / No. Gradients

Test	hi-fi eval	lo-fi eval	total t	factor
PORT with hi-fi analyses, 2 var	14/13		$\approx$ 12 hrs	
AMMO, 2 var	3/3	19/9	$\approx$ 2.41hrs	$\approx$ 5
PORT with hi-fi analyses, 84 var	19/19		$\approx$ 35 hrs	
AMMO, 84 var	4/4	23/8	$\approx$ 7.2hrs	$\approx$ 5

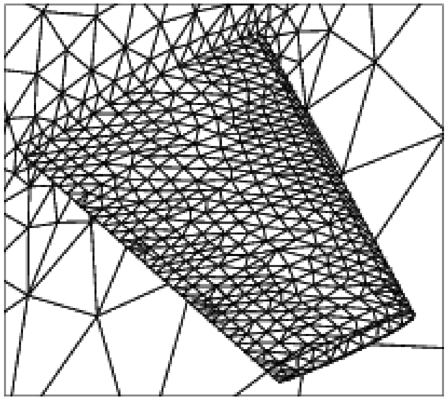
$C_D^{initial}$ = 0.0171 at α=1°, flap y-displacement=0
C_D^{final} = 0.0148 at α=1.6305°, flap y-displacement=-0.0048
a decrease of $\approx$ 13.45%

3D Aerodynamic Design with AMMO

Hi-fi: FUN3D N-S on a finer mesh



Lo-fi: FUN3D Euler on a coarse mesh



$$\min_x \quad 5C_D^2 + \tfrac{1}{2}(C_L - 0.12303)^2$$
$$s.t. \quad x_l \le x \le x_u$$

α_0=3.06°, M_∞=0.84, Re=5x10^6

Lift$_0$ = 0.12302, Drag$_0$ = 0.01713, Objective$_0$ = 0.0014670

Cost Reduction with AMMO
(No. functions / No. gradients)

Test	Hi-fi eval	Lo-fi eval	Final Lift	Final Drag	f
PORT/hi-fi	13/11		0.11146	0.01532	0.0012793
AMMO	3/3	22/15	0.10657	0.01511	0.0012796

- **Factor 2 savings in terms of wall-clock time**
- **Further savings are expected upon development of optimal termination criteria for low-fidelity subproblem computations**
- **Large-scale 3D slot wing design in progress**

Work in Progress

- **Computational expense is still a difficulty**
  - Investigating optimal termination of the low-fidelity computations based of sufficient predicted decrease
  - Investigating MASSOUD as a potential robust and efficient volume grid manipulation tool
  - Choice of "optimal" models: least expensive, but with good predictive properties
- **Explicit constraint handling in optimization problems**
  - Complex derivatives
  - Adjoints when variables outnumber responses
- **Handling mesh adaptation or regenerating meshes in optimization**
- **Robust handling of analysis and mesh movement failure**

Part II:

MDO Problem Synthesis and Solution

Background

- **MDO formulation**
  - **Statement of the problem as a nonlinear program (subset of the total design problem)**
- **Optimization algorithm**
  - **Scheme for solving the formulation**
- **Analytical features of MDO problem formulation strongly influence the practical ability of optimization algorithms to solve the MDO problem reliably and efficiently**
- **Can not afford *ad hoc* techniques with computationally intensive models**

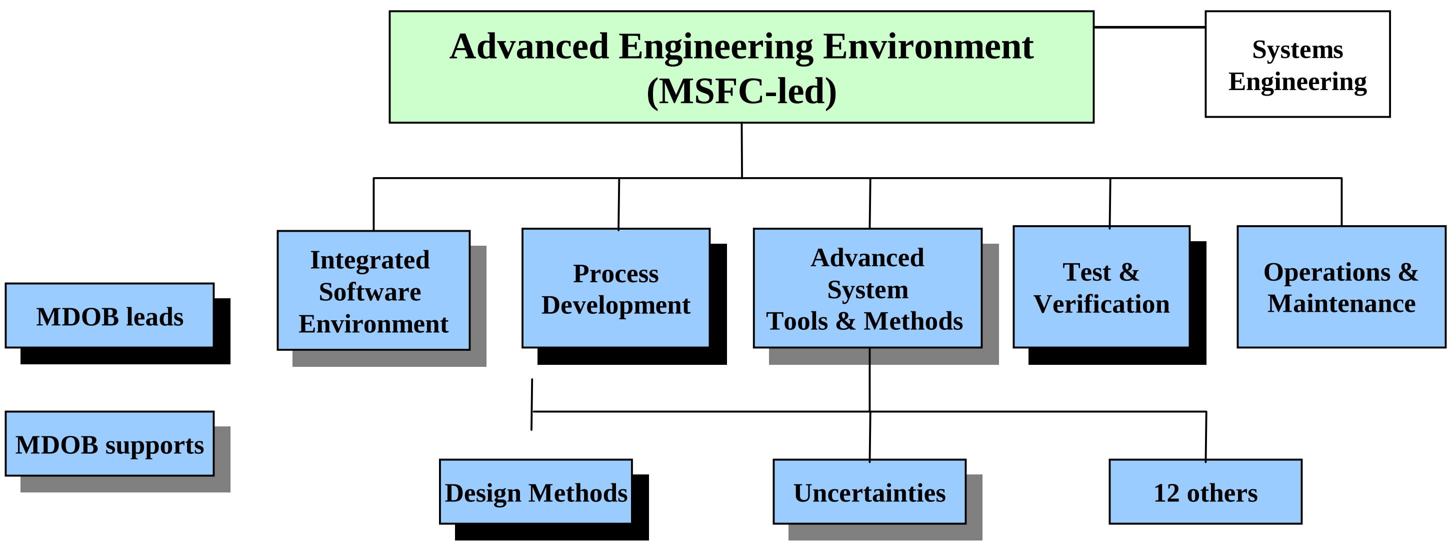
Background

- **Goal**
  - Provide robust MDO problem formulations and attendant algorithms to ensure efficient and reliable solution strategies for the design problem
- **Work done in some form since MDOB founded**
- **Chronological connection to programs**
  - Base (FY95-97)
  - HPCCP/HSCT (FY97-00)
  - ASCoT (FY01)
  - 2^{nd} Gen RLV/AEE (FY02)
- **Collaborators:**

FY95-present	N.M. Alexandrov (MDOB)	Analysis and algorithms
FY99-present	R.M. Lewis (ICASE, W & M)	Analysis
FY98	S. Kodiyalam (Engineous, Inc.)	Comparative computational study
		of problem formulations
FY01	P.L. Shepherd (W & M student)	Computational interfaces

2nd Generation RLV Project
(2001-2002)

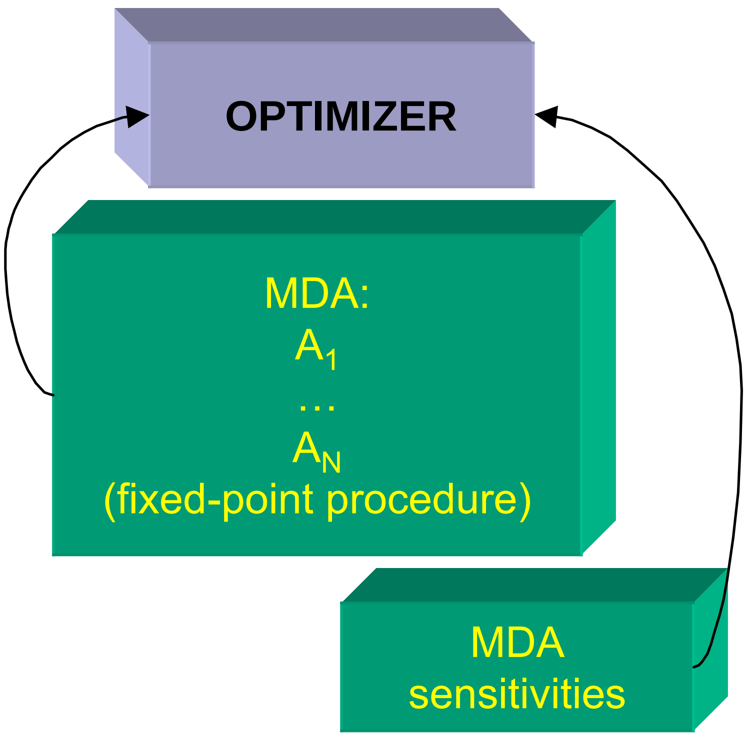
- **NASA's goals for the second generation RLV are to:**
  - Improve the expected safety of launch so that by the year 2010 the probability of losing a crew is no worse than 1 in 10,000 missions.
  - Reduce the cost of delivering a pound of payload to low Earth orbit from today's $10,000 down to $1000 by the year 2010.
- **2nd Gen RLV/AEE Objective**
  - Deliver to the 2nd Generation RLV Program and ISAT Team "an advanced engineering synthesis environment complete with life-cycle simulation models capable of modeling technology, uncertainty, cost and risk"



Canonical MDO Problem Synthesis: Fully Integrated Formulation (FIO)

Problem: **design for objective f with**

i = 1, …, N
and constraints

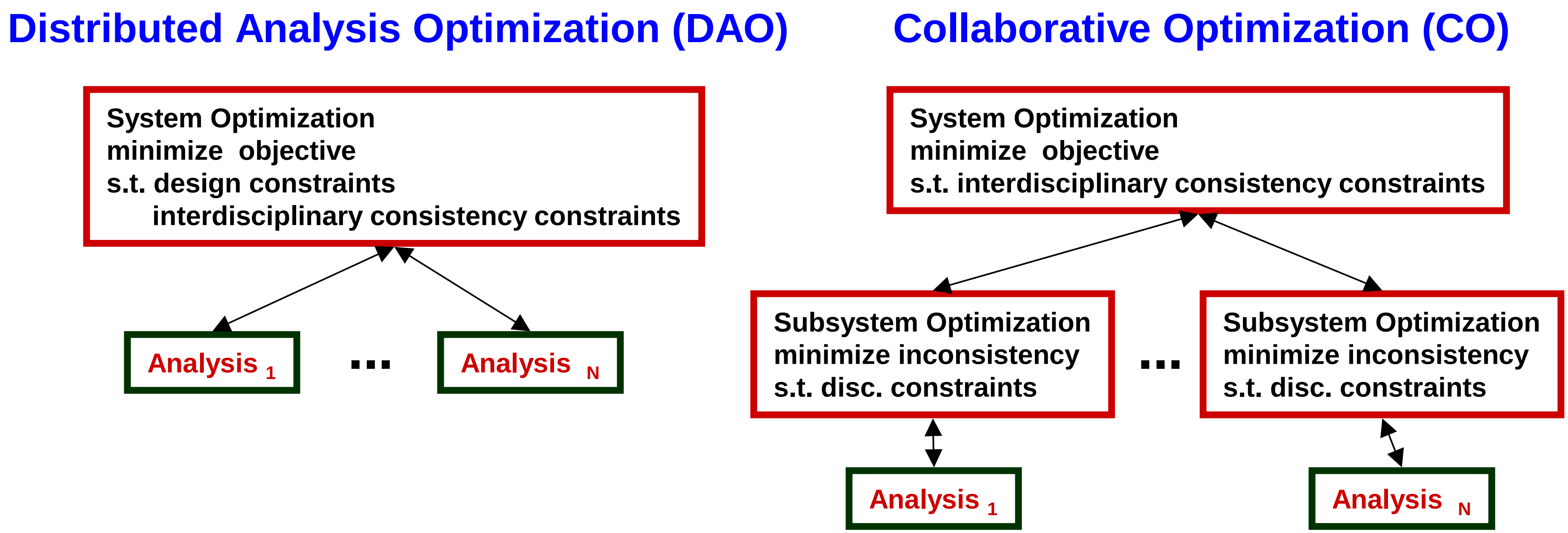


- **Laborious, expensive, one-time process**
- **Difficult to transform or expand**
- **Need to develop MDA-based derivatives**
- **Assumes that MDA is done via fixed-point iteration**
- **Expensive to maintain MDA far from solution**
- **Little disciplinary autonomy**
- **Drawbacks of FIO motivate other formulations**

HPCCP/HSCT Formulation Study

Alexandrov and Kodiyalam, AIAA-98-4884

- Dramatic differences in performance
- Formulations in the study: FIO and



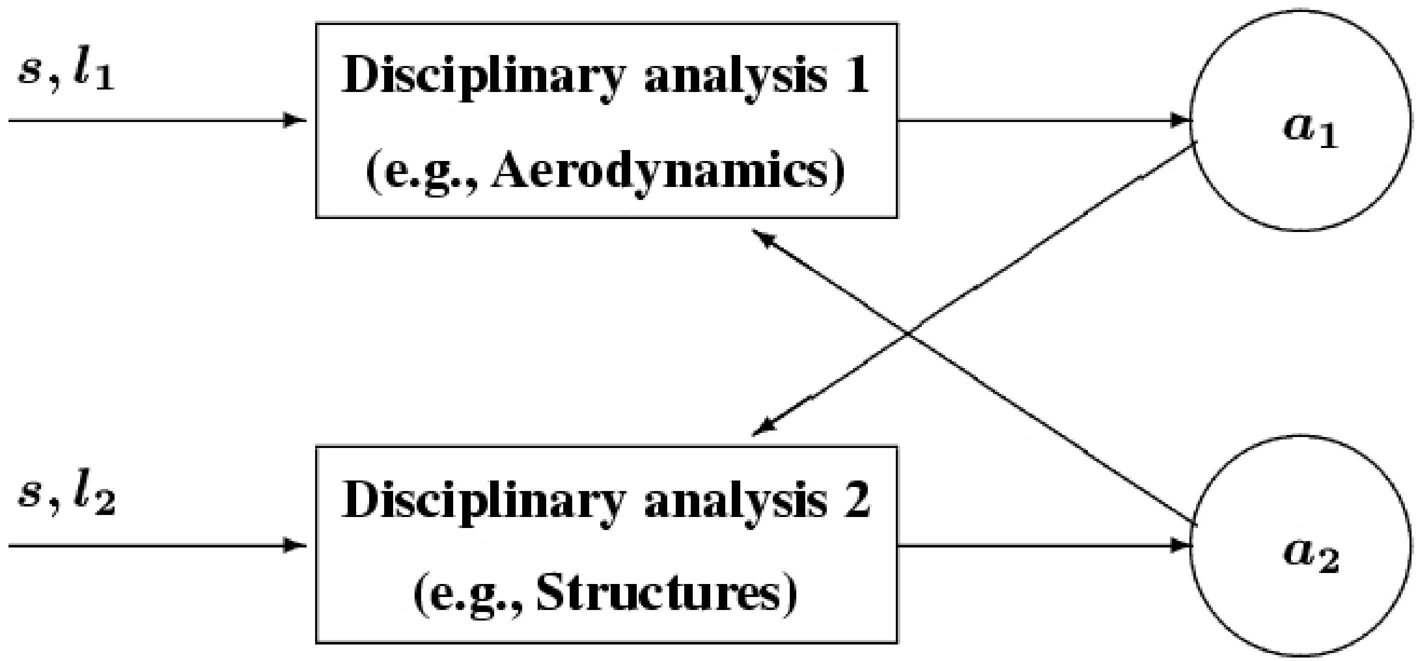
Problem / Method	1	2	3	4	5	6	7	8	9	10
MDF	610	220	610	81	3234	5024	8730	245	1574	1353
CO	15626	19872	1785	2102	837	40125	691058	–	–	–
IDF	9530	8976	382	–	544	932	–	–	–	–

Example: representative # analyses (MDF = FIO, IDF$\subset$DAO)

Evaluating a Formulation

- **Amenable to solution?**
- **Robust formulation?**
  - Is the solution set the same as that of the canonical problem?
  - Do answers satisfy necessary conditions?
  - Is it sensitive to small changes in parameters?
- **Efficiency of solution?**
- **Autonomy of implementation / ease of transformation?**
  - Claim: this is the most labor-intensive part
  - Important because no single formulation is good for all problems
- **Autonomy of execution?**
  - Wish to follow organizational structure for design
  - Wish to optimize wrt local variables only in disciplines
- **These questions are important in practice**
  - Direct influence on software and solubility

The Two-Discipline Model Problem



- Coupled MDA $\sim$ the physical requirement that a solution satisfy both analyses
- Given $x = (s, l_1, l_2)$, we have

$$
\begin{aligned}
a_1 &= A_1(s, l_1, a_2) \\
a_2 &= A_2(s, l_2, a_1)
\end{aligned}
$$

Relationship among Optimization Problem Formulations

Write MDA as
$$
\begin{aligned}
a_1 &= A_1(s, l_1, t_2) \\
a_2 &= A_2(s, l_2, t_1) \\
t_1 &= a_1 \\
t_2 &= a_2
\end{aligned}
$$

Start with Simultaneous Analysis and Design (SAND) formulation:

$$
\begin{aligned}
\underset{s, a_1, a_2, l_1, l_2, t_1, t_2}{\text{minimize}} \quad & f_{SAND}(s, a_1, a_2) \\
\text{subject to} \quad & g_1(s, l_1, a_1) \geq 0 \\
& g_2(s, l_2, a_2) \geq 0 \\
& a_1 = A_1(s, l_1, t_2) \\
& a_2 = A_2(s, l_2, t_1) \\
& t_1 = a_1 \\
& t_2 = a_2
\end{aligned}
$$

Relationship among Optimization Problem Formulations (cont.)

- **Eliminate subsets of variables from SAND by *closing* various subsets of constraints $\Longrightarrow$ get other formulations:**
  - **Distributed Analysis Optimization (DAO):** Eliminate a_1, a_2 as independent variables by closing the disciplinary analysis constraints at every iteration of optimization
  - **Fully Integrated Optimization (FIO):** In addition, eliminate t_1, t_2 as independent variables by closing $t_1 = a_1$ and $t_2 = a_2$.
  - **Optimization by Linear Decomposition (OLD):** Eliminate l_1, l_2, t_1, t_2 as independent variables via optimization subproblems (MDA remains)
  - **Collaborative Optimization (CO):** Eliminate l_1, l_2 (but not t_1, t_2) via optimization subproblems

Autonomy / Modularity in Implementation

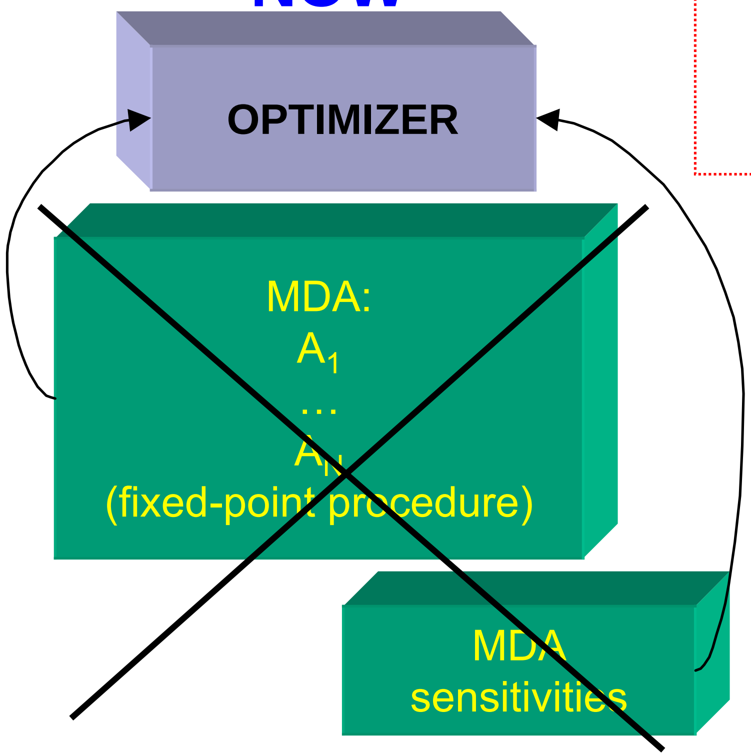
- **Computational elements needed for optimization (in particular, sensitivities) can be implemented autonomously by disciplines**
- **All formulations require roughly the same amount of work to implement**
- **Can reconfigure the same set of computational components to implement one formulation of another**

MDO Problem Synthesis / Implementation

Problem: **design for objective f with**

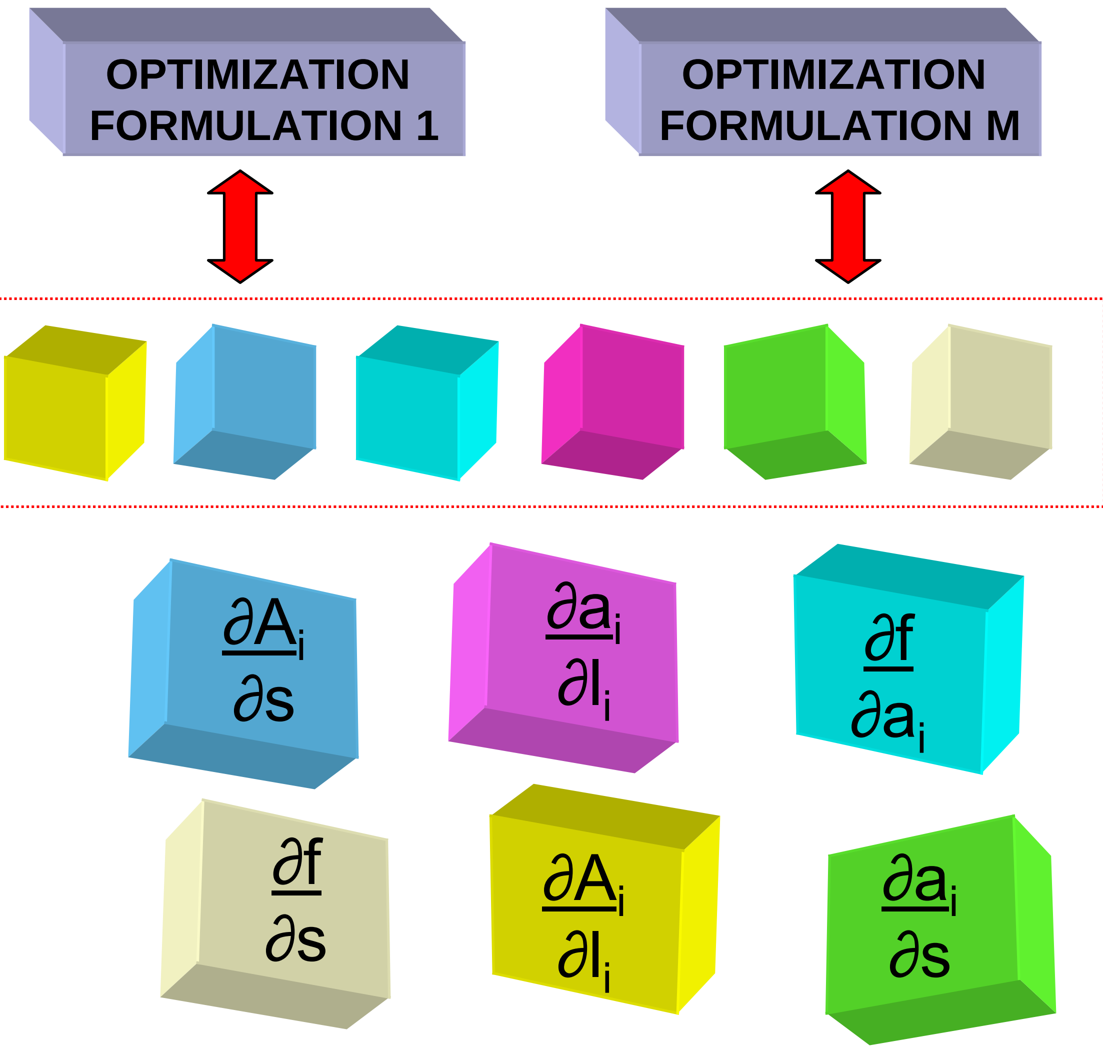
i = 1, …, N

NOW



Laborious, expensive, one-time integration, difficult to transform/ expand

FUTURE



Expend the effort at the outset to implement analysis and sensitivity modules; easy to transform and expand: an opportunity for a general framework

Example: Sensitivities in DAO vs FIO

Consider DAO:

$$\begin{aligned}
\underset{s,l_1,l_2,t_1,t_2}{\textbf{minimize}} \quad & f_{DAO}(s,t_1,t_2) = f(s, a_1(s,l_1,l_2,t_2), a_2(s,l_1,l_2,t_1)) \\
\textbf{subject to} \quad & g_0(s,t_1,t_2) \geq 0 \\
& g_1(s,l_1,t_1) \geq 0 \\
& g_2(s,l_2,t_2) \geq 0 \\
& t_1 = a_1(s,l_1,l_2,t_2) \\
& t_2 = a_2(s,l_2,l_2,t_1),
\end{aligned}$$

where, given $(s,l_1,l_2,t_1,t_2), a_1$ and a_2 are found from

$$\begin{aligned}
a_1 - A_1(s,l_1,t_2) &= 0 \\
a_2 - A_2(s,l_2,t_1) &= 0.
\end{aligned}$$

Example: Sensitivities in DAO vs FIO, cont.

For the objective $f_{DAO}(s, t_1, t_2)$, we need

$$\frac{\partial f}{\partial s}, \frac{\partial f}{\partial t_1}, \frac{\partial f}{\partial t_1}$$

For the design constraints $g_1(s, l_1, t_1)$ and $g_2(s, l_2, t_2)$ we need

$$\frac{\partial g_1}{\partial s}, \frac{\partial g_1}{\partial l_1}, \frac{\partial g_1}{\partial t_1} \quad \text{and} \quad \frac{\partial g_2}{\partial s}, \frac{\partial g_2}{\partial l_2}, \frac{\partial g_2}{\partial t_2}.$$

For the consistency constraints $t_1 - A_1(s, l_1, t_2) = 0$ and

$t_2 - A_2(s, l_2, t_1) = 0$ we need

$$\frac{\partial A_1}{\partial s}, \frac{\partial A_1}{\partial l_1}, \frac{\partial A_1}{\partial t_2} \quad \text{and} \quad \frac{\partial A_2}{\partial s}, \frac{\partial A_2}{\partial l_2}, \frac{\partial A_2}{\partial t_1}.$$

Example: Sensitivities in DAO vs FIO, cont.

Consider FIO:

$$
\begin{array}{ll}
\underset{s,l_1,l_2}{\textbf{minimize}} & f(s, a_1(s, l_1, l_2), a_2(s, l_1, l_2)) \\
\textbf{subject to} & g_0(s, l_1, a_1(s, l_1, l_2), a_2(s, l_1, l_2)) \geq 0 \\
& g_1(s, l_1, a_1(s, l_1, l_2)) \geq 0 \\
& g_2(s, l_2, a_2(s, l_1, l_2)) \geq 0,
\end{array}
$$

where a_1 and a_2 are computed in MDA

$$
\begin{aligned}
a_1 &= A_1(s, l_1, a_2) \\
a_2 &= A_2(s, l_2, a_1)
\end{aligned}
$$

Example: Sensitivities in DAO vs FIO, cont.

In FIO approach, we need to compute the sensitivities of the objective

$$f_{FIO}(s, l_1, l_2) = f(s, a_1(s, l_1, l_2), a_2(s, l_1, l_2)).$$

By the chain rule,

$$\begin{aligned}
\frac{\partial f_{FIO}}{\partial s} &= \frac{\partial f}{\partial s} + \frac{\partial f}{\partial a_1}\frac{\partial a_1}{\partial s} + \frac{\partial f}{\partial a_2}\frac{\partial a_2}{\partial s} \\
\frac{\partial f_{FIO}}{\partial l_1} &= \frac{\partial f}{\partial a_1}\frac{\partial a_1}{\partial l_1} + \frac{\partial f}{\partial a_2}\frac{\partial a_2}{\partial l_1} \\
\frac{\partial f_{FIO}}{\partial l_2} &= \frac{\partial f}{\partial a_1}\frac{\partial a_1}{\partial l_2} + \frac{\partial f}{\partial a_2}\frac{\partial a_2}{\partial l_2}
\end{aligned}$$

We compute the derivatives of a_1 and a_2 by implicit differentiation of the multidisciplinary analysis equations

$$\begin{aligned}
a_1 - A_1(s, l_1, a_2) &= 0 \\
a_2 - A_2(s, l_2, a_1) &= 0
\end{aligned}$$

This yields

$$\begin{pmatrix} I & -\dfrac{\partial A_1}{\partial a_2} \\ -\dfrac{\partial A_2}{\partial a_1} & I \end{pmatrix} \begin{pmatrix} \dfrac{\partial a_1}{\partial s} \\ \dfrac{\partial a_2}{\partial s} \end{pmatrix} = - \begin{pmatrix} \dfrac{\partial A_1}{\partial s} \\ \dfrac{\partial A_2}{\partial s} \end{pmatrix},$$

$$\begin{pmatrix} I & -\dfrac{\partial A_1}{\partial a_2} \\ -\dfrac{\partial A_2}{\partial a_1} & I \end{pmatrix} \begin{pmatrix} \dfrac{\partial a_1}{\partial l_1} \\ \dfrac{\partial a_2}{\partial l_1} \end{pmatrix} = - \begin{pmatrix} \dfrac{\partial A_1}{\partial l_1} \\ 0 \end{pmatrix},$$

and

$$\begin{pmatrix} I & -\dfrac{\partial A_1}{\partial a_2} \\ -\dfrac{\partial A_2}{\partial a_1} & I \end{pmatrix} \begin{pmatrix} \dfrac{\partial a_1}{\partial l_2} \\ \dfrac{\partial a_2}{\partial l_2} \end{pmatrix} = - \begin{pmatrix} 0 \\ \dfrac{\partial A_2}{\partial l_2} \end{pmatrix}$$

to be solved for the sensitivities of a_1 and a_2 wrt (s, l_1, l_2). (Referred to as the "generalized sensitivity equations" by Sobieski, 1990)

Example: Sensitivities in DAO vs FIO, cont.

- **Observe that the same elements are needed for FIO and DAO sensitivity computations**
- **Can implement constituent elements with disciplinary autonomy if** ***do not integrate MDA via fixed-point iteration early***
- **The elements are integrated differently in FIO and DAO**
- **Analogous results for CO and OLD**
- **Conclusion: The same computational components are required**

Algorithmic Interactions

- **Saw how, in principle, can re-arrange computational components associated with one formulation and obtain components for another**
- **Re-arrangement may require substantial effort**
- **Now show how for some of the formulations, minor changes in an optimization algorithm may yield an algorithm for solving another formulation**
- **Straightforward to pass among some formulations $\Longrightarrow$ facilitate the use of hybrid approaches: may use one far from solution, another near solution**

Example: DAO vs FIO vs SAND (analysis and coupling constraints only)

Simplified FIO formulation: $\underset{x}{\text{minimize}} \quad f_{FIO}(x) \equiv f(x, a_1(x), a_2(x)),$

where, given x, we solve the MDA

$$\begin{pmatrix} \tilde{A}_1(x) \\ \tilde{A}_2(x) \end{pmatrix} = \begin{pmatrix} a_1 - A_1(x, a_1(x), a_2(x)) \\ a_2 - A_2(x, a_1(x), a_2(x)) \end{pmatrix} = 0$$

Simplified SAND formulation:

$$\begin{array}{ll} \underset{x, a_1, a_2}{\text{minimize}} & f_{SAND}(x, a_1, a_2) \equiv f(x, a_1, a_2) \\ \text{subject to} & \tilde{A}_1(x, a_1, a_2) = 0 \\ & \tilde{A}_2(x, a_1, a_2) = 0 \end{array}$$

Simplified DAO formulation:

$$\begin{array}{ll} \underset{x, a_1, a_2, t_1, t_2}{\text{minimize}} & f_{DAO}(x, a_1, a_2) \\ \text{subject to} & t_1 - a_1(x, t_1, t_2) = 0 \\ & t_2 - a_2(x, t_1, t_2) = 0 \end{array}$$

Example: DAO vs FIO vs SAND, cont.

W_i — basis of the null-space associated with the derivative of the block A_i. Relying on implicit differentiation and the derivations by Lewis, 1997, note the relationship among the sensitivities for the three methods:

- Suppose, (x, a) is feasible with respect to MDA. Then the (projected) gradients at (x, a) of FIO and SAND are related by

$$\nabla_x f_{FIO}(x) = W_{SAND}^T(x, a)\nabla_{x,a} f_{SAND}(x, a),$$

  where W_{SAND} denotes a particular basis for the null-space of $\nabla\tilde{A}^T$ in the SAND approach.

- Suppose that (x, a) is feasible with respect to MDA. Then

$$W_{DAO}^T \nabla_{x,a} f_{DAO}(x, a) = W_{SAND}^T(x, a)\nabla_{x,a} f_{SAND}(x, a)$$

Can use these relationships to implement a reduced-basis optimization algorithm for the three formulations with minimal modifications.

Sketch of a conceptual algorithm

Consider one step of a reduced-basis algorithm for the SAND formulation:

1. Construct a local model of the Lagrangian about the current design.
2. Take a substep to improve feasibility.
3. Subject to improved feasibility, take a substep to improve optimality.
4. Set the total step to the sum of the substeps, evaluate and update.

- MDA after step 4 $\Longrightarrow$ a corresponding algorithm for FIO.
- Solving the disciplinary equations as in DAO $\Longrightarrow$ an algorithm for DAO.
- Passing between algorithms for distinct formulations is a straightforward step.

Our Currently Favorite Formulation: Expanded DAO

$$
\begin{aligned}
\underset{s,\sigma_0,\sigma_1,\sigma_2,l_1,l_2,t_1,t_2}{\text{minimize}} \quad & f_{DAO}(s, t_1, t_2) \\
\text{subject to} \quad & g_0(\sigma_0, t_1, t_2) \geq 0 \\
& g_1(\sigma_1, l_1, t_1) \geq 0 \\
& g_2(\sigma_2, l_2, t_2) \geq 0 \\
& t_1 = a_1(\sigma_1, l_1, t_2) \\
& t_2 = a_2(\sigma_2, l_2, t_1) \\
& \sigma_0 = s \\
& \sigma_1 = s \\
& \sigma_2 = s
\end{aligned}
$$

- Expand variable space to relax the requirement that the disciplinary design constraints be satisfied with the system-level values of s
- Implementation autonomy, no MDA
- Single-level optimization problem - readily soluble

Moral of the Story

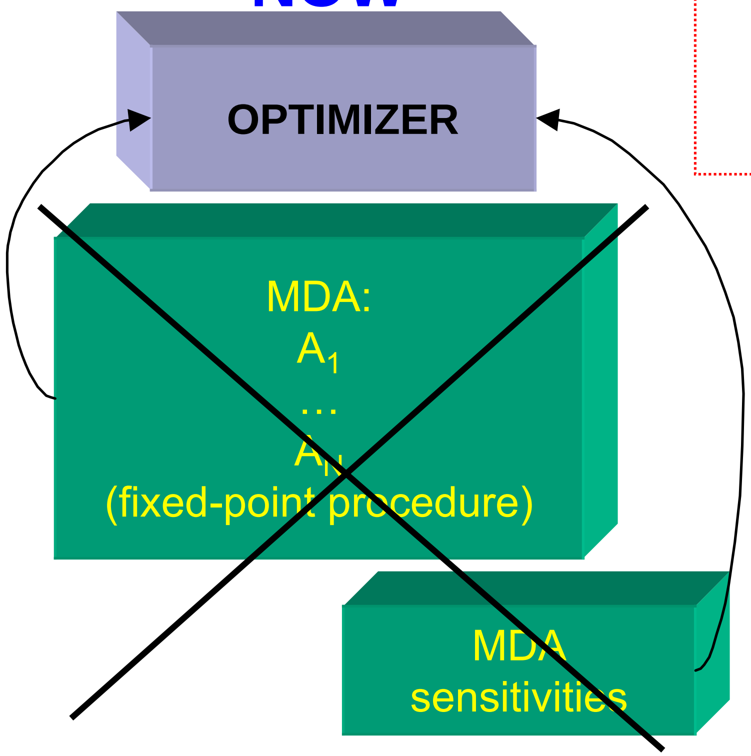
- **Problem formulation determines the practical solubility of the MDO problem**
- **No single formulation or algorithm is good for all problems**
- **Need to ease implementation of the formulations and enable easy interchange among formulations and hybrid formulations**
- **All formulations need roughly the same components – identify them**
- **Create disciplinary modules that can be reconfigured dynamically**

MDO Problem Synthesis / Implementation

Problem: design for objective f with

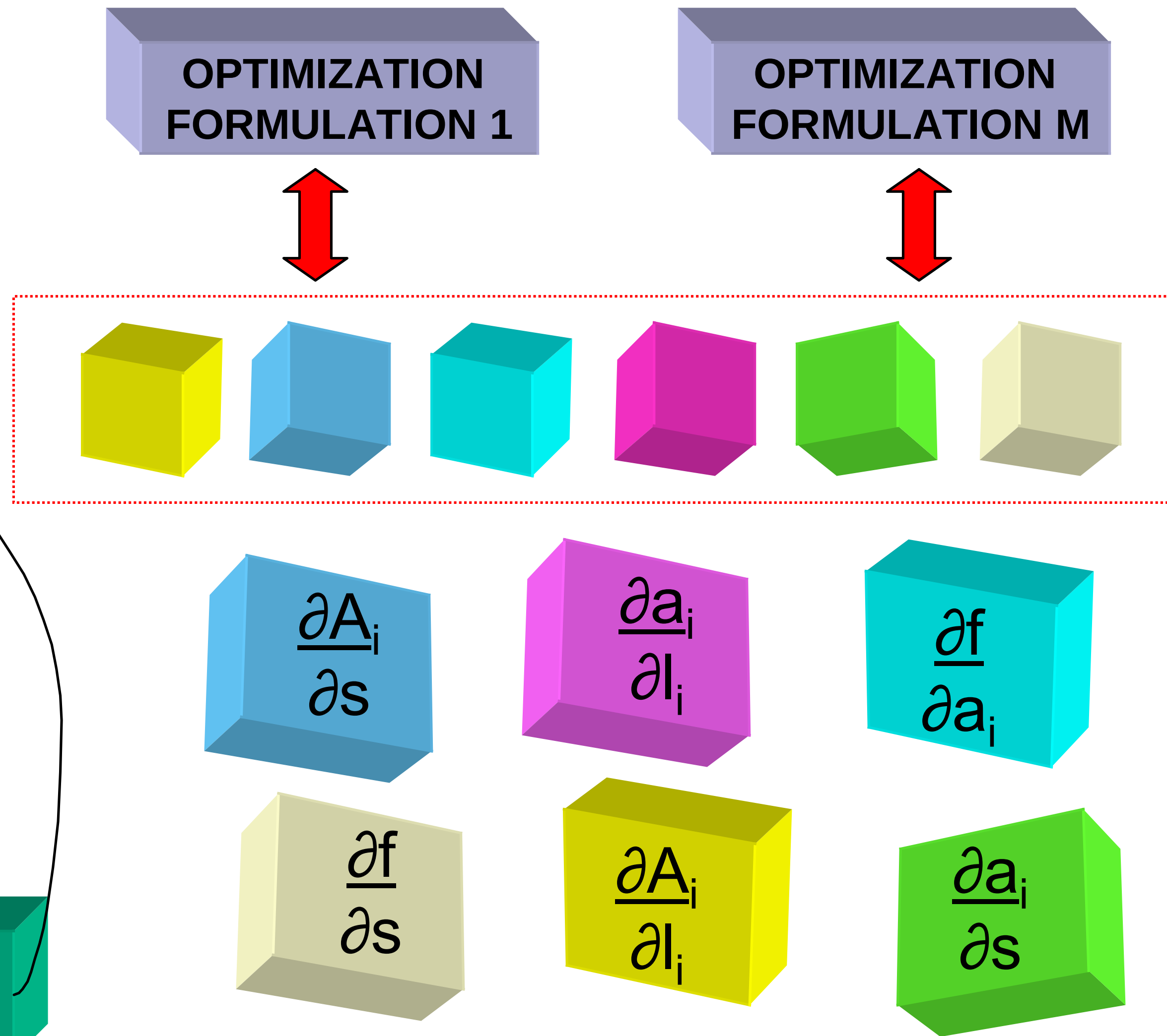
i = 1, …, N

NOW



Laborious, expensive, one-time integration, difficult to transform/ expand

FUTURE



Expend the effort at the outset to implement analysis and sensitivity modules; easy to transform and expand: an opportunity for a general framework

Work in Progress

- Identification of modular computational components in the context of several distributed formulations and attendant algorithms
- Evaluation of competing distributed formulations and optimization strategies
- Multiobjective optimization for analysis-based design
- Robust multilevel strategies
- Current work in the context of Model Center (Phoenix Integration) and Dakota (Sandia Labs) frameworks
- Deliverables to 2nd Gen RLV/AEE:
  - Specifications for a modular optimization framework
  - Demonstrations

Appendix: Comparative Summary of Formulations

- **FIO:** Single-level optimization, arbitrary coupling, some autonomy of implementation, MDA required
- **SAND:** Single-level optimization, arbitrary coupling, some autonomy of implementation, MDA not done, large optimization problem
- **DAO:** Single-level optimization, not for broadly coupled problems, autonomy of implementation, some autonomy of execution
- **CO:** Bilevel optimization, autonomy of implementation and autonomy of execution (distributed MDA), local variables handled in subproblems, no MDA, not for broadly coupled problems, not robust, can be difficult to solve
- **OLD:** Bilevel optimization, MDA required, autonomy of implementation and some autonomy of execution, not robust, can be difficult to solve

Model Management References

- Alexandrov, N. M.; Lewis, R. M.: "First-Order Model Management for Engineering Optimization", Optimization and Engineering, 2001, in press.
- Alexandrov, N. M.; Lewis, R. M.: "First-Order Approximation and Model Management in Optimization", Large-Scale PDE-Constrained Optimization, 2001, Springer-Verlag, Berlin, in press.
- Alexandrov, N. M.; Nielsen, E. J.; Lewis, R. M.; Anderson, W. K.: "First-Order Model Management with Variable-Fidelity Physics Applied to Multi-Element Airfoil Optimization", AIAA Paper 2000-4886, also Journal of Aircraft, in press.
- Alexandrov, N. M.; Lewis, R. M.; Gumbert, C. R.; Green, L. L.; and Newman, P.A.: "Optimization with Variable-Fidelity Models Applied to Wing", AIAA Paper 2000-0841.
- Alexandrov, N.: "On Managing the Use of Surrogates in General Nonlinear Optimization and MDO", AIAA Paper 99-4798.
- Alexandrov, N.: "A Trust-Region Framework for Managing Approximations in Constrained Optimization and MDO Problems", ISSMO/NASA 1st Internet Conference on Approximations and Fast Re-Analysis in Engineering Optimization, June 14-27, 1998.

MDO Synthesis References

- Alexandrov, N. M.; Lewis, R. M.: "First-Order Model Management for Engineering Optimization", Optimization and Engineering, 2001, in press.
- Alexandrov, N. M.; Lewis, R. M.: "Analytical and Computational Properties of Collaborative Optimization", NASA/TM-210104-2000, AIAA Journal, in press.
- Alexandrov, N. M.; Lewis, R. M.: "Analytical and Computational Properties of Distributed Approaches to MDO", AIAA Paper 2000-4719, 2000.
- Alexandrov, N. M.; Lewis, R. M.: "Algorithmic Perspectives on Problem Formulation in MDO", AIAA Paper 2000-4718, 2000.
- Alexandrov, N.; and Kodiyalam, S.: "Initial Results of an MDO Method Evaluation Study," AIAA Paper 98-4884, 1998.