Problem Synthesis in MDO: an Overview

Natalia M. Alexandrov

Multidisciplinary Optimization Branch
NASA Langley Research Center
Hampton, Virginia

http://mdob.larc.nasa.gov

n.alexandrov@larc.nasa.gov

Outline

- Components of MDO
- MDO problem synthesis
  - Problem synthesis study
  - A two-discipline model problem
  - MDO problem formulations and their properties
  - Modular implementation
  - Algorithmic interactions
- Managing simulation-based models
- Concluding remarks

Multidisciplinary Design Optimization

- MDO – systematic approaches to the design of complex, coupled systems
- “Multidisciplinary” – different aspects of the design problem
- Actual definition depends on application, stage of design, etc.
- Define the MDO problem as the subset of the total design problem that can be expressed as a nonlinear programming problem.

Some Defining Features of MDO Problems

- Complexity of constituent analyses
- Difficulty of component integration
- Computational expense of function and constraint evaluations
- The need to attain multidisciplinary equilibrium at solutions
- Multiobjective nature of the problem
- Unreliable (non-automatic) evaluation of functions and constraints
- …

Computational Components of MDO

Design-Oriented Analysis ✓	Design Problem Synthesis and Solution ✓	Computational Infrastructure
Variable-fidelity models	Design problem formulation with analysis	Analysis frameworks
Data-fitting approximations	Decomposition and synthesis strategies	Design optimization frameworks
Error estimates and bounds	SD/MD optimization algorithms, including multilevel optimization	Data standards
Uncertainty quantification	Managing variable-fidelity models in optimization	Software engineering
MD analysis	Nontraditional methods	Data/process visualization
Sensitivity analysis	Multiobjective optimization and decision making	User interaction/expert-in-the-loop
Automation and robustness	Optimization under uncertainty	Integration

A Component of MDO: Problem Synthesis

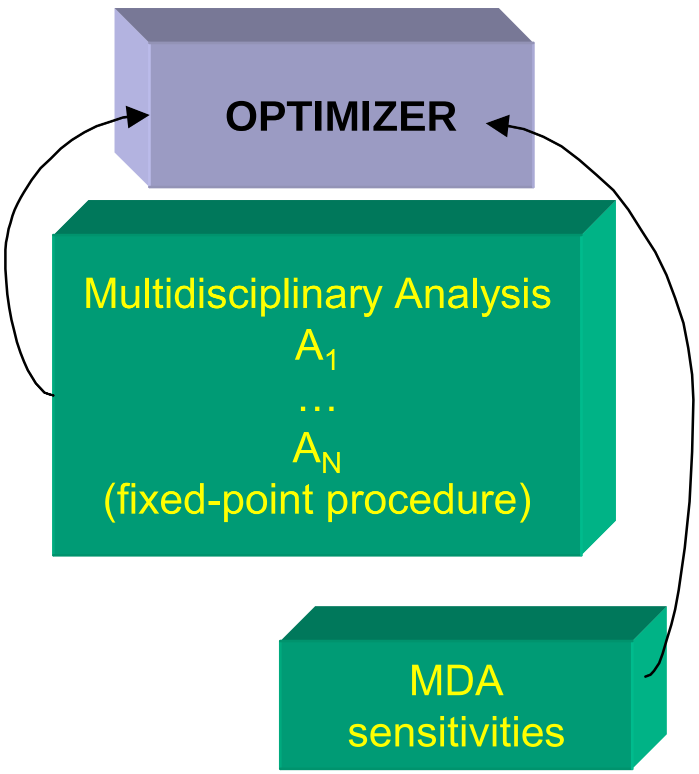
Background

- **MDO formulation**
  - Statement of the problem as a nonlinear program
    - Analysis answers questions of equivalence to canonical formulation, well-posedness, optimality conditions, solubility, sensitivity of solutions to perturbations in parameters
- **Optimization algorithm**
  - Scheme for solving the formulation
    - Analysis answers questions of global convergence, local convergence rates, etc.
- **Analytical features of MDO problem formulation strongly influence the practical ability of optimization algorithms to solve the MDO problem reliably and efficiently**

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 Multidisciplinary Analysis (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**

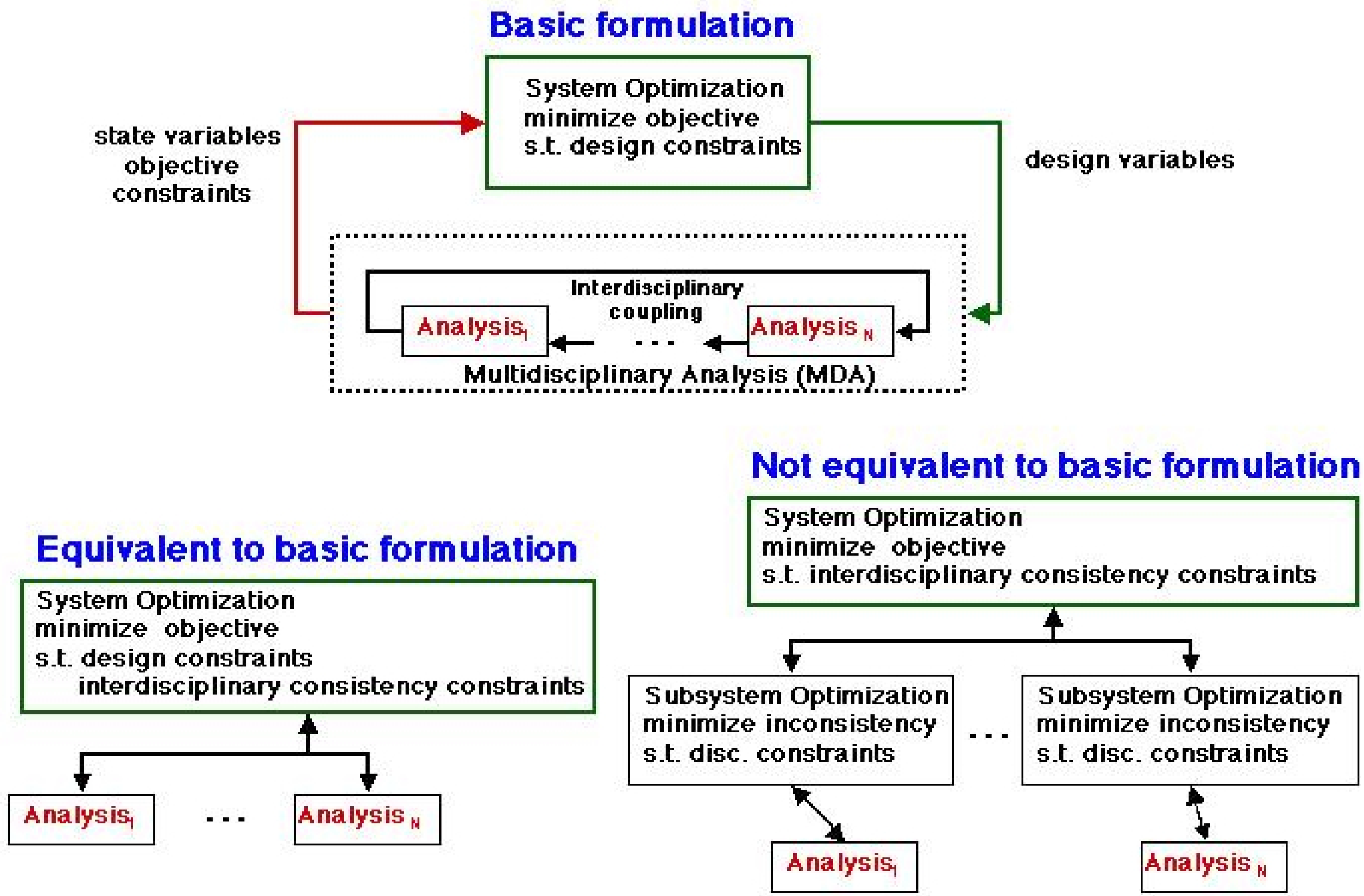
Motivation for Analysis and Study

- Most alternatives to FIO are based on *ad hoc* approaches
- Anecdotal evidence indicates that some methods work better than others
- Limited computational evidence of relative performance properties (next page)
- Mathematical analysis looks into reasons

Example: HPCCP/HSCT Formulation Study

Alexandrov and Kodiyalam, AIAA-98-4884

Evaluated 3 formulations with respect to a number of performance metrics:



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

Example, continued

- Contributing formulations
  - Basic formulation (FIO)
  - Equivalent (Distributed Analysis Optimization, DAO)
  - Non-equivalent (Collaborative Optimization, CO)
- Dramatic differences in performance

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)

Basic formulation for a two-discipline problem (simplified)

$$\begin{aligned}
\underset{s,l_1,l_2}{\text{minimize}} \quad & f\Big(s, R_1(u_1(s,l_1)), R_2(u_2(s,l_2))\Big) \\
\text{subject to} \quad & g_1(s, l_1, u_1(s,l_1)) \geq 0 \\
& g_2(s, l_2, u_2(s,l_2)) \geq 0,
\end{aligned}$$

where, given (s, l_1, l_2), (u_1, u_2) is the solution of the MDA

$$\begin{aligned}
A_1\Big(s, l_1, u_1(s,l_1), T_1(u_2(s,l_2))\Big) &= 0 \\
A_2\Big(s, l_2, u_2(s,l_2), T_2(u_1(s,l_1))\Big) &= 0
\end{aligned}$$

s - shared variables, l_i - local variables, R_i, T_i - variable transformations.

- Amenable to standard NLP algorithms
- The smallest optimization problem
- Can be efficient and may be necessary
- MDA difficult to implement and expensive to use

An equivalent formulation

$$\begin{array}{ll}\underset{s,l_1,l_2,u_{12},u_{21}}{\text{minimize}} & f\Big(s, R_1\big(u_1(s,l_1,u_{12})\big), R_2\big(u_2(s,l_2,u_{21})\big)\Big) \\ \text{subject to} & g_1\big(s,l_1,u_1(s,l_1,u_{12})\big) \ge 0 \\ & g_2\big(s,l_2,u_2(s,l_2,u_{21})\big) \ge 0 \\ & u_{12} - T_1\big(u_2(s,l_2,u_{21})\big) = 0 \\ & u_{21} - T_2\big(u_1(s,l_1,u_{12})\big) = 0,\end{array}$$

where, given $(s, l_1, l_2, u_{12}, u_{21})$, u_1 and u_2 are solutions of independent

$$\begin{aligned} A_1\big(s, l_1, u_1(s,l_1,u_{12}), u_{12}\big) &= 0 \\ A_2\big(s, l_2, u_2(s,l_2,u_{21}), u_{21}\big) &= 0. \end{aligned}$$

- Retains analytic properties of the basic formulation
- MDA attained at solution, not at every iteration
- A larger optimization problem

A non-equivalent formulation (a CO_2 version)

$$\begin{array}{ll} \underset{s,u_{12},u_{21}}{\textbf{minimize}} & f\Big(s, R_1\big(u_1(s,u_{12})\big), R_2\big(u_2(s,u_{21})\big)\Big) \\ \textbf{subject to} & c_1(s,u_{12}) = \| \sigma_1 - s \|^2 + \| T_1(u_2) - u_{12} \|^2 \\ & c_2(s,u_{21}) = \| \sigma_2 - s \|^2 + \| T_2(u_1) - u_{21} \|^2 \end{array}$$

c_i - interdisciplinary consistency constraints

$\sigma_i(s,u_{ij})$ $l_i(s,u_{ij})$ are computed by

$$\begin{array}{ll} \underset{\sigma_i,l_i}{\textbf{minimize}} & \| \sigma_i - s \|^2 + \| T_i(u_j(\sigma_i,l_i)) - u_{ij} \|^2 \\ \textbf{subject to} & g_i\big(\sigma_i, l_i, u_i(\sigma_i,l_i)\big) \geq 0 \end{array}$$

In the disciplinary subproblems u_i are computed via

$$A_i\big(\sigma_i, l_i, u_i(\sigma_i, l_i, u_{ij}), u_{ij}\big) = 0$$

Salient characteristics of the non-equivalent formulation

- **Solution set is equivalent to that of the basic formulation**
- **MDA is not attained until solution**
- **Nonlinear, nonconvex, bilevel programming problem**
- **Features that will cause difficulties for optimization algorithms (and exist even if the functions of the basic formulation are perfectly well behaved):**
  - **System-level constraints make it difficult to find feasible points**
  - **System-level constraints may be, in a practical sense, discontinuous**
  - **Lagrange multipliers do not exist for the system-level problem**
  - **Optimization problems will be more nonlinear than the original problem**
  - **Derivatives of system-level constraints (CO_1) will be discontinuous**
  - **The difficulties occur at and near solutions of the system-level problem**

Illustration: World's simplest problem

(e.g., a bar of fixed length and variable cross-section area under a longitudinal force)

$$\text{minimize}\{s \mid 0 \le s \le 1\}$$

On reformulating as CO_2, system and subsystem problems become

$$\begin{array}{ll} \underset{s}{\text{minimize}} & f(s) \\ \text{subject to} & c_1(s) = \tfrac{1}{2} \| s - \sigma_1(s) \|^2 = 0 \\ & c_2(s) = \tfrac{1}{2} \| s - \sigma_2(s) \|^2 = 0 \end{array}$$

$$\min\{\frac{1}{2} \| \sigma_1 - s \|^2 \mid \sigma_1 \ge 0\} \text{ and } \min\{\frac{1}{2} \| \sigma_2 - s \|^2 \mid \sigma_2 \le 1\}$$

One readily checks that the subproblem solutions are

$$\sigma_1(s) = \begin{cases} 0 & \text{if } s \le 0 \\ s & \text{if } s \ge 0 \end{cases} \qquad \sigma_2(s) = \begin{cases} s & \text{if } s \le 1 \\ 1 & \text{if } s \ge 1 \end{cases}$$

Example continued

Breakdown of the standard stationarity conditions in CO_2

- $\nabla c_i(s) = s - \sigma_i(s)$ and at $s_* = \alpha, \nabla c_1(s_*) = 0$
- Stationarity conditions: there exist λ_1 and λ_2 such that

$$\nabla f(s_*) + \lambda_1 \nabla c_1(s_*) + \lambda_2 \nabla c_2(s_*) = 0$$

- But $\nabla f(s_*) + \lambda_1 \nabla c_1(s_*) + \lambda_2 \nabla c_2(s_*) = \nabla f(s_*) = 1$
- Algorithms rely on the stationarity conditions for
  - computing steps
  - gauging progress
  - making decisions about termination
- Could start at a solution and not recognize it

Example continued:

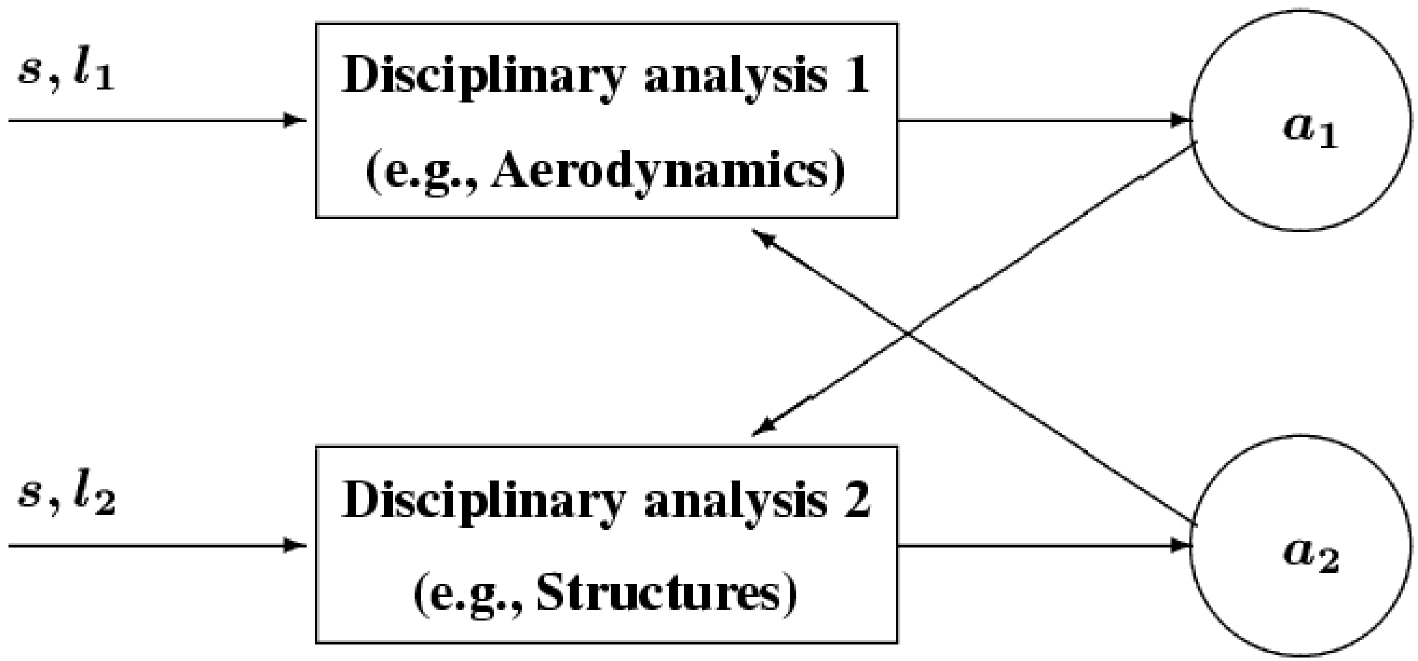
Results of NPSOL with
$s_0 = 0.001$ and
$s_* = 0$

Iteration	s	Penalty
0	1.000e-03	0.0e+00
1	-9.990e-01	4.2e+00
2	-9.847e-01	5.7e+00
3	-8.282e-01	7.4e+00
4	-4.142e-01	2.7e+01
5	-3.430e-01	5.9e+01
6	-1.718e-01	4.0e+02
7	-1.436e-01	8.2e+02
8	-7.251e-02	5.4e+03
9	-6.076e-02	1.1e+04
10	-3.203e-02	6.5e+04
11	-2.717e-02	1.2e+05
12	-1.727e-02	5.1e+05
13	-1.442e-02	1.9e+06
14	-1.414e-02	4.7e+06

Intermediate summary

- **Formulations are distinguished from algorithms**
- **Formulations are equivalent if**
  - **Solutions sets are equivalent**
  - **Algorithmic implications are similar**
- **Reformulating a problem can make it much harder to solve**
- **Some objectives can be accomplished by an algorithm – no need to complicate the problem formulation**
- **Coupling must be resolved somewhere**
- **If avoiding MDA is the goal, can use an equivalent alternative to the basic formulation**

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

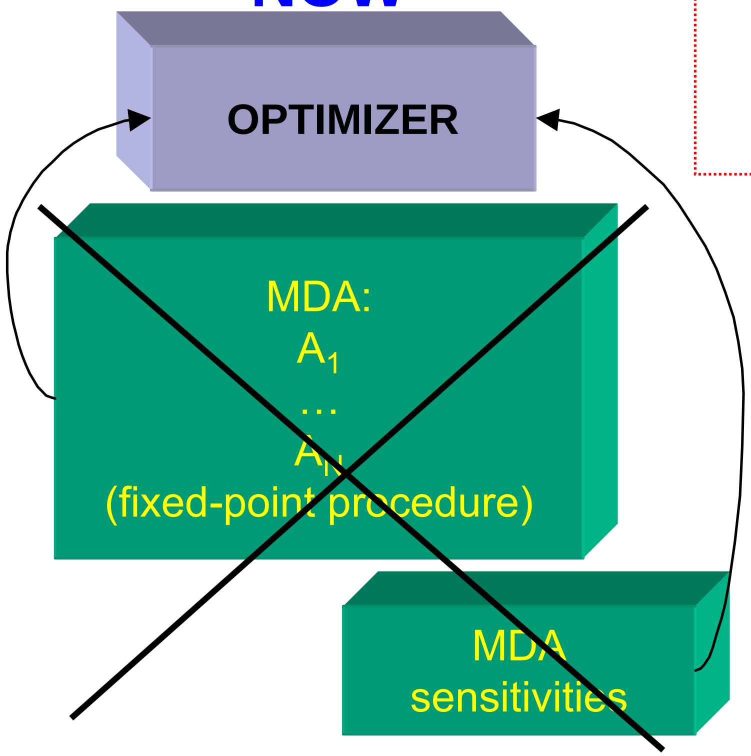
- Computational elements needed for optimization (in particular, sensitivities) can be implemented autonomously by disciplines
- All formulations require roughly the same amount of work to implement
- Can reconfigure the same set of computational components to implement one formulation of another

MDO Problem Synthesis / Implementation

Problem: design for objective f with

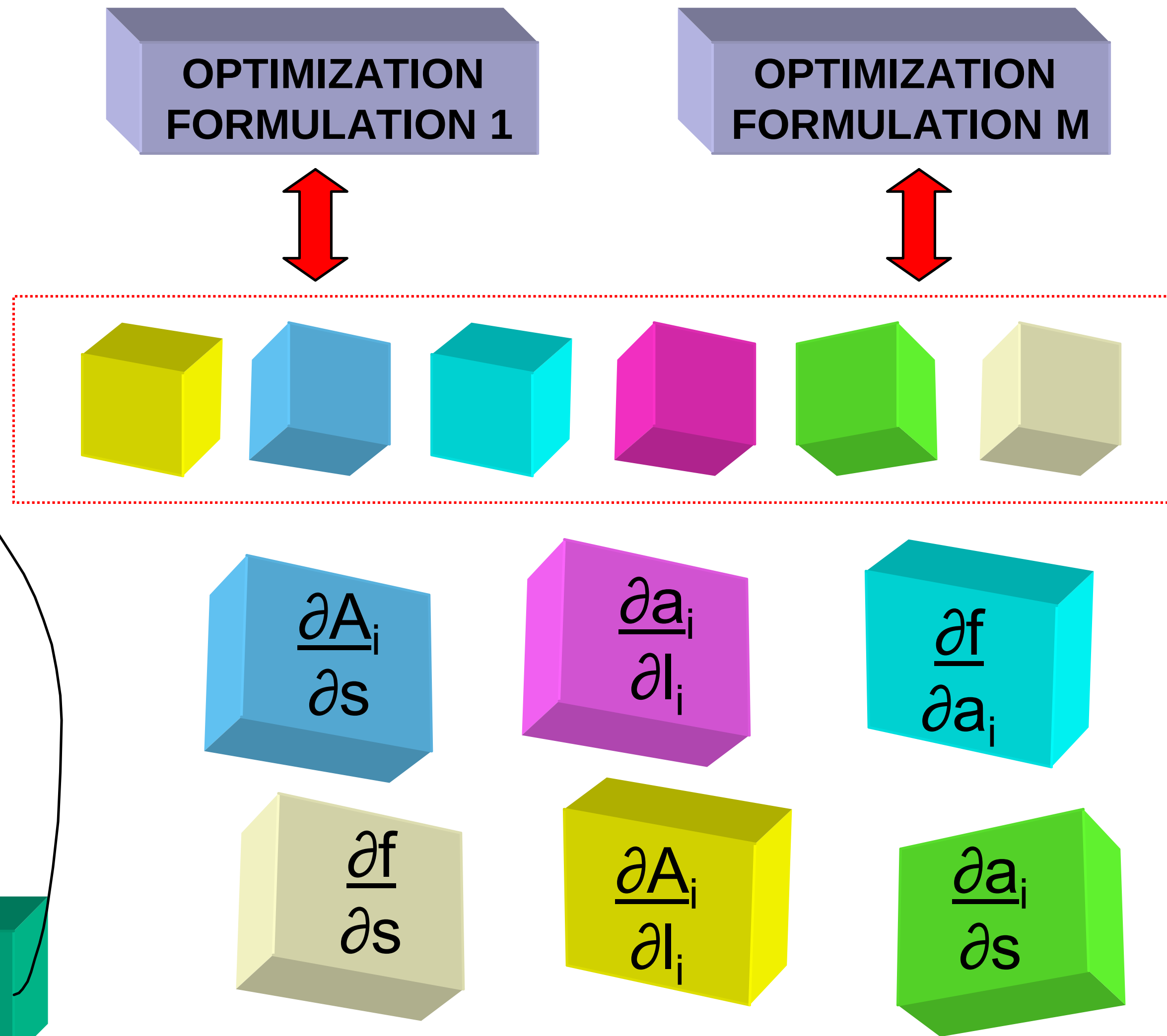
i = 1, …, N

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:

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

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

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

- **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

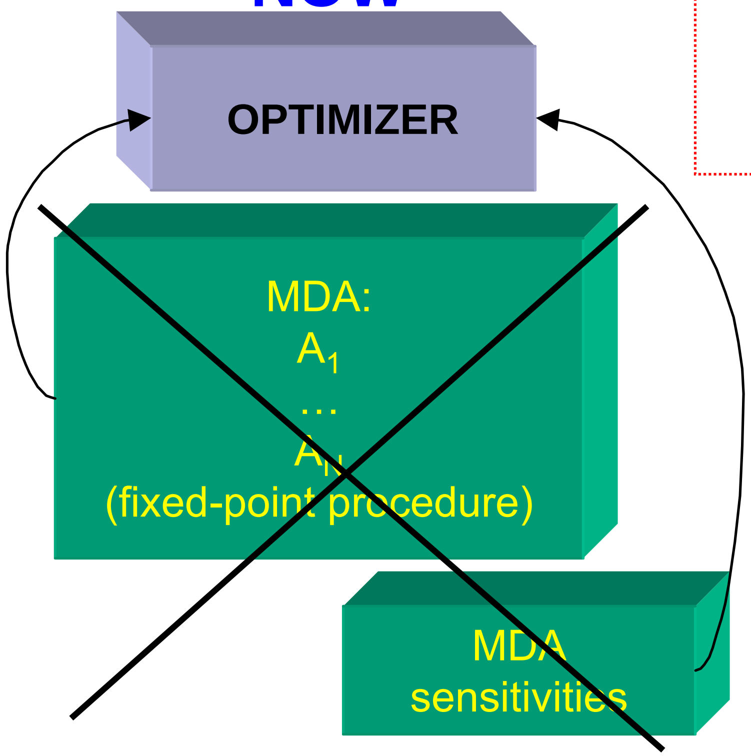
- **Problem formulation determines the practical solubility of the MDO problem**
- **No single formulation or algorithm is good for all problems**
- **Need to ease implementation of the formulations and enable easy interchange among formulations and hybrid formulations**
- **All formulations need roughly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

MDO Problem Synthesis / Implementation

Problem: **design for objective f with**

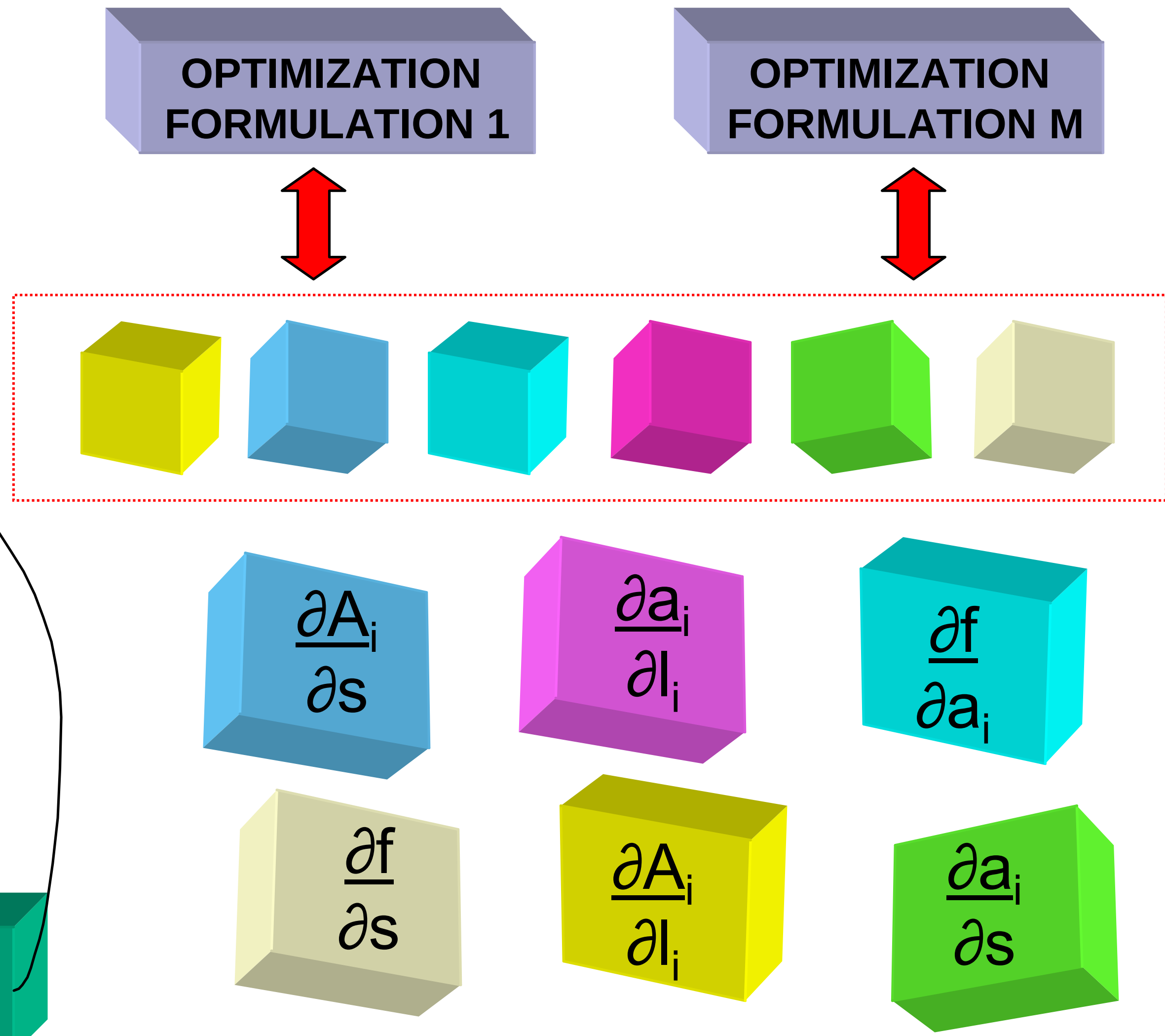
i = 1, …, N

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

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

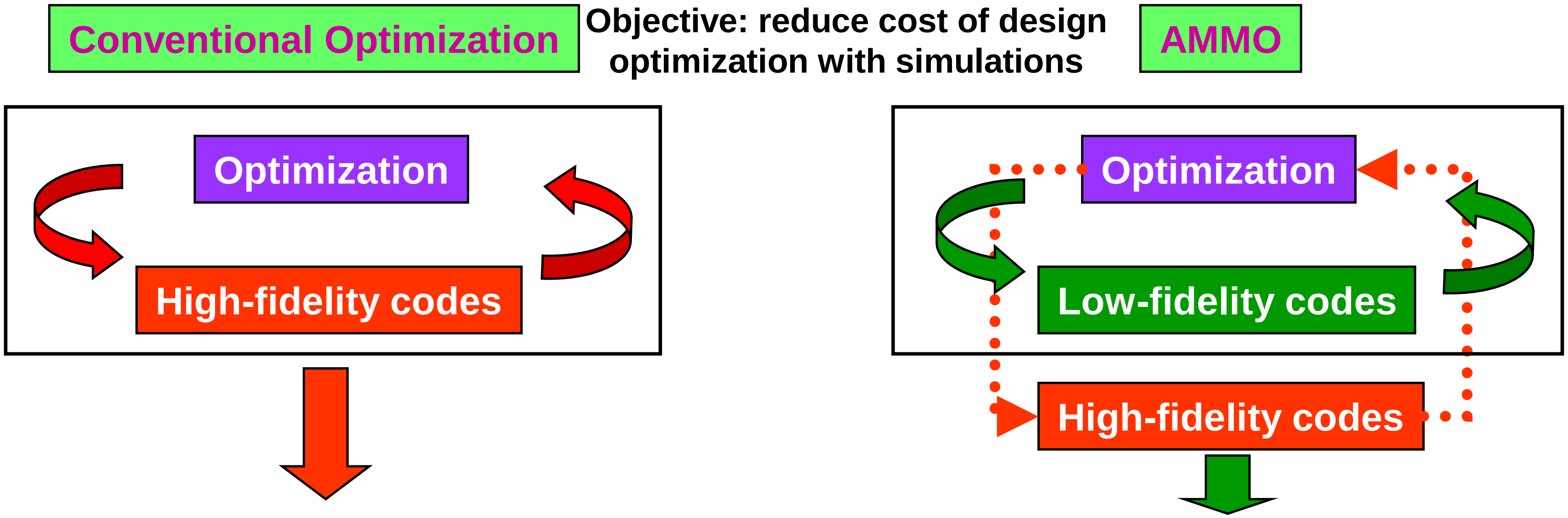
Managing simulation-based models

- **Limiting factors:**
  - Extreme expense of repeated simulations
  - Function and derivative evaluations prone to failure away from the nominal design
  - Derivative-free optimization is not an option due to computational expense

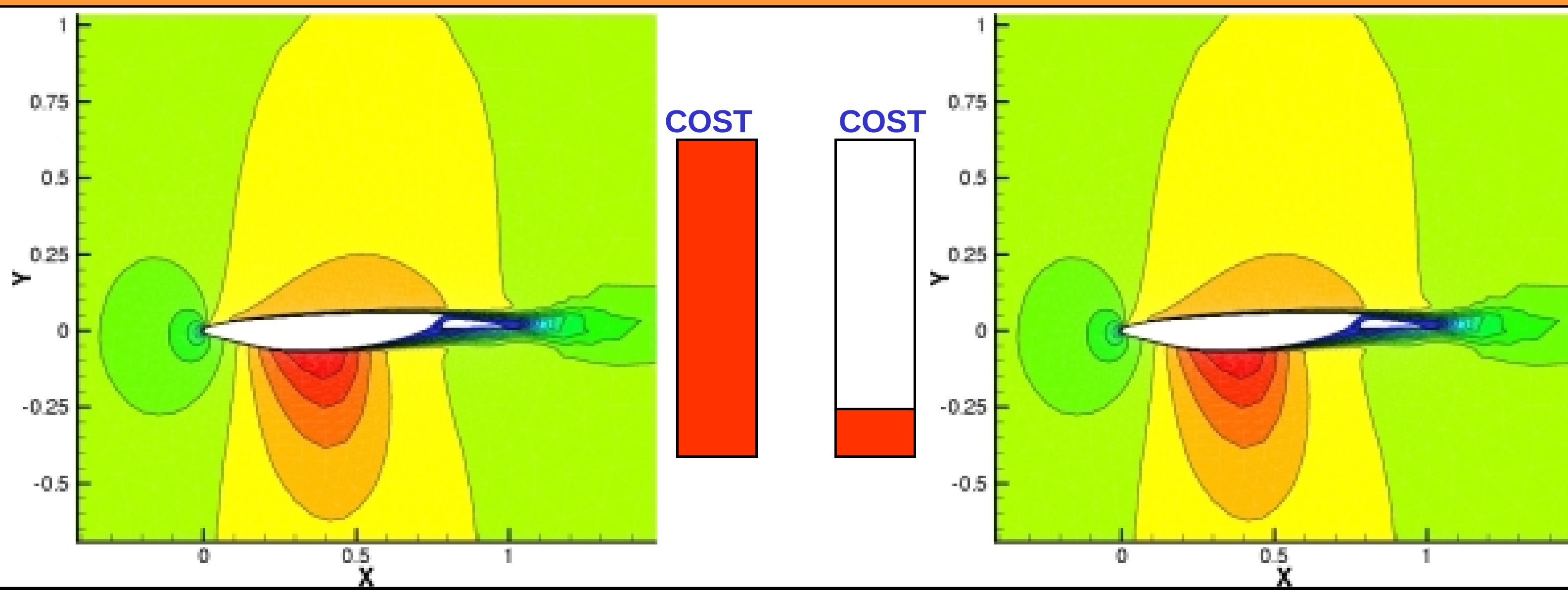
Approach

- **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



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



AMMO gives Navier-Stokes answers with five-fold savings

Concluding Remarks

- MDO is a very complex problem
  - Synthesis is difficult
  - Function evaluations are not automated
  - Infrastructure is in its infancy
- Some current promising areas
  - Modeling for design optimization
  - Rigorous approaches to problem synthesis

Some Publications on MDO Problem Synthesis:

Alexandrov, N. M.; Lewis, R. M.: "Analytical and Computational Properties of Collaborative Optimization", NASA/TM-210104-2000, AIAA Journal, in press. LTRS

Alexandrov, N. M.; Lewis, R. M.: "Analytical and Computational Properties of Distributed Approaches to MDO", AIAA Paper 2000-4719, 8th AIAA/USAF/NASA/ISSMO Symposium on MA&O, Long Beach, CA, 9/5-9/00. LTRS

Alexandrov, N. M.; Lewis, R. M.: "Algorithmic Perspectives on Problem Formulation in MDO", AIAA Paper 2000-4718, 8th AIAA/USAF/NASA/ISSMO Symposium on MA&O, Long Beach, CA, 9/5-9/00. LTRS

Alexandrov, N.; and Kodiyalam, S.: "Initial Results of an MDO Method Evaluation Study," AIAA Paper 98-4884, 7th AIAA/USAF/NASA/ISSMO Symposium on Multidisciplinary Analysis and Optimization, St. Louis, MO, Sept. 2-4, 1998. LTRS

Some Publications on Model Management:

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-Constrainted 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, 8th AIAA/USAF/NASA/ISSMO Symposium on Multidisciplinary Analysis and Optimization, Long Beach, CA, 6-8 Sept. 2000; also Journal of Aircraft, in press. LTRS.

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, 38th Aerospace Sciences Meeting and Exhibit, 10-13 January 2000, Reno, NV. LTRS.

Alexandrov, N.: "On Managing the Use of Surrogates in General Nonlinear Optimization and MDO", AIAA Paper 99-4798, 7th AIAA/USAF/NASA/ISSMO Symposium on Multidisciplinary Analysis and Optimization, St. Louis, MO, Sept. 2-4, 1998. LTRS

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.