MULTIdisciplinary Multiobjective Problem Synthesis and Optimization

Natalia Alexandrov

Multidisciplinary Optimization Branch
NASA Langley Research Center
Hampton, VA

http://fmad-www.larc.nasa.gov/mdob/MDOB

Outline

- Introduction to MULTI
  - Objective
  - Where MULTI fits within the LCS element
  - Component definitions
  - Overview of the components
- Some work to-date: Problem formulation for MDO
  - Motivation
  - Algorithmic perspective on MDO problem formulation
  - Summary
- Major open issues
- FY01 plans

Objective

- Enable robust and rigorous design optimization of large, multidisciplinary engineering systems on distributed computer networks at all stages of life-cycle design, from system studies to preliminary to detailed design.

Where MULTI Fits within the LCS Element

- "The life-cycle tool box will develop simulation-based, variable-fidelity decision and analysis tools with emphasis on cost, risk and probabilistic modeling. These tools … will enable the advancement of engineering practices through the modeling of life-cycle processes, advanced optimization techniques and robust design methods."

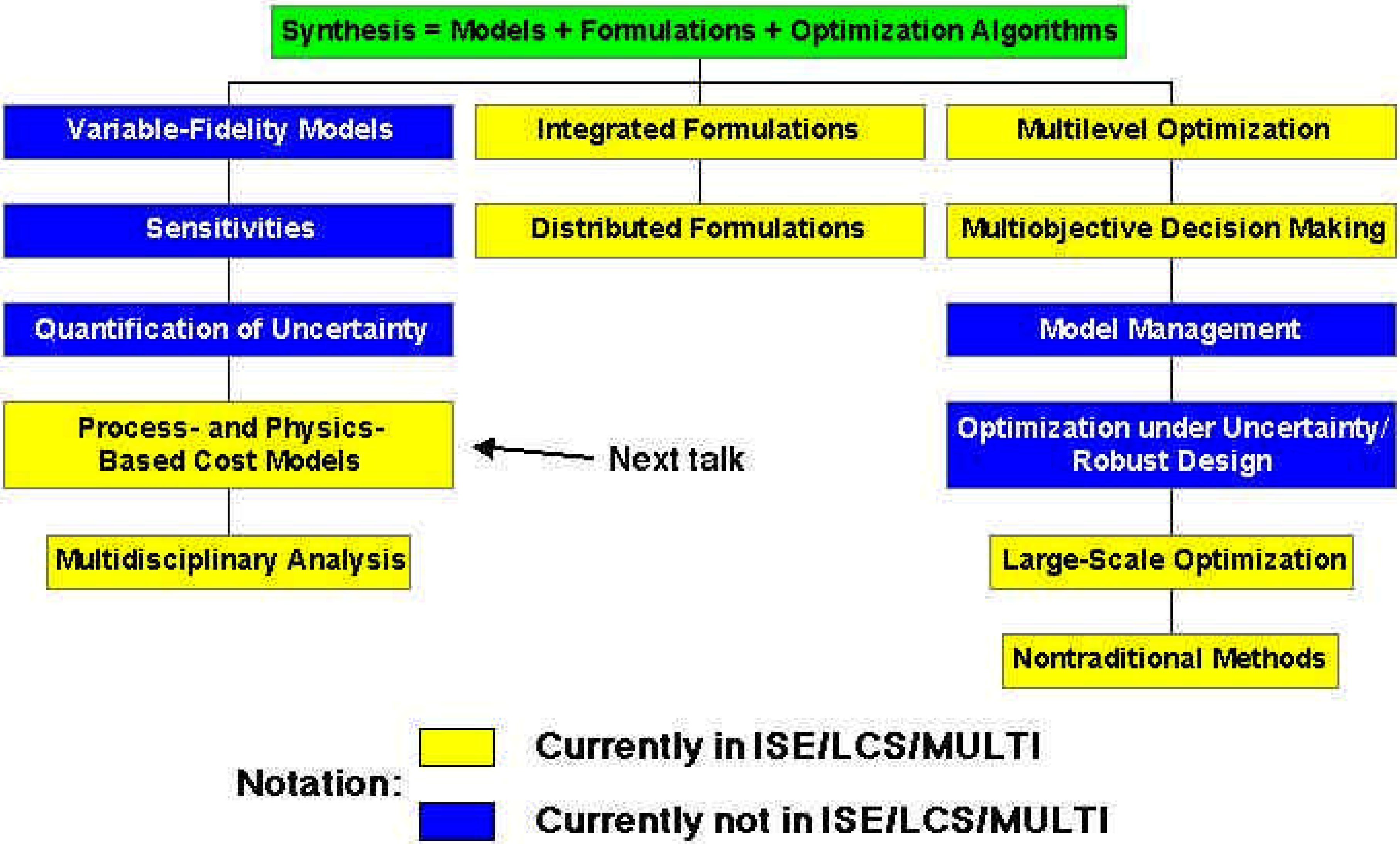
ISE Program plan, Sept. 2000

Defintions:

Synthesis = Models + Formulations + Optimization Algorithms

- **Models**
  - Variable-fidelity models from disciplines:
    - Variable-fidelity physics, variable-resolution, variable-accuracy, reduced order, reduced basis…
  - Data-fitting models
- **Problem formulations**
  - Statements of the design problem as a mathematical programming problem amenable to solution – single-level, bilevel, or multilevel
- **Optimization algorithms**
  - Recipes for solving problem formulations translated into software

MULTI "Toolbox": Overview of Components



Models and Model Management for Design

(component overview continued)

- **Objective**
  - Enable design optimization with simulations by reducing the cost of using high-fidelity models in MDO
- **Approach**
  - Place the computational load on lower-fidelity models while maintaining convergence to high-fidelity results
- **Components**
  - Building variable-fidelity models for design
  - Quantifying uncertainty in models and associated sensitivities
  - Managing variable-fidelity models in optimization
- **Results to-date** (under ASCOT/FAAST)
  - For optimization with variable-fidelity model management, fivefold savings in terms of high-fidelity analyses and sensitivities (MDOB) compared to conventional design
  - Rigorous bounds on outputs of PDE (MIT)

Problem Formulation for MDO
(component overview continued)

- **Objective**
  - Develop and demonstrate provably robust MDO problem formulations with approaches to problem decomposition, synthesis, and subsystem autonomy
- **Approach**
  - Identify critical modeling and optimization requirements
  - Analyze existing promising MDO problem formulations
  - Develop robust problem formulations
  - Develop guidelines for the use of formulations
- **Results to-date**
  - Later in the talk

Optimization Algorithms for Design
(component overview continued)

- **Objective**
  - Develop and demonstrate optimization methods, including multilevel methods, for large-scale, multiobjective, distributed optimization problem formulations
- **Results to-date**
  - Proposed a provably convergent multilevel method for large-scale optimization
  - Several home-grown and commercial alternatives for multiobjective optimization are under consideration
  - Demonstration problem identified – multiobjective cost-performance optimization of an RLV

Some work to-date: Problem Formulation for MDO

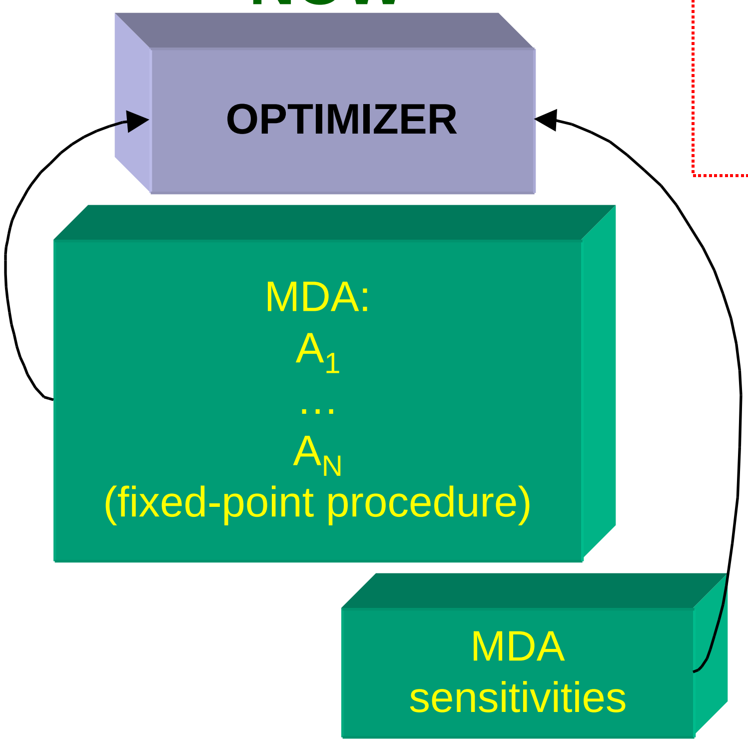
- **Motivation**
  - Analytical properties of MDO problem formulations have a direct and powerful influence on the practical solution of the resulting computational optimization problem
  - Difficulties may be introduced by attempting to attain desirable organizational and structural goals
  - Current state of the art: “one-of-a-kind”, laborious and time consuming MDO problem formulation and implementation
- **Goal**
  - Robust, modular, “interchangeable” formulations that can be reliably and efficiently solved by existing optimization algorithms

MDO Synthesis: One-of-a-Kind vs. Modular, Reusable

Problem: design for objective f with

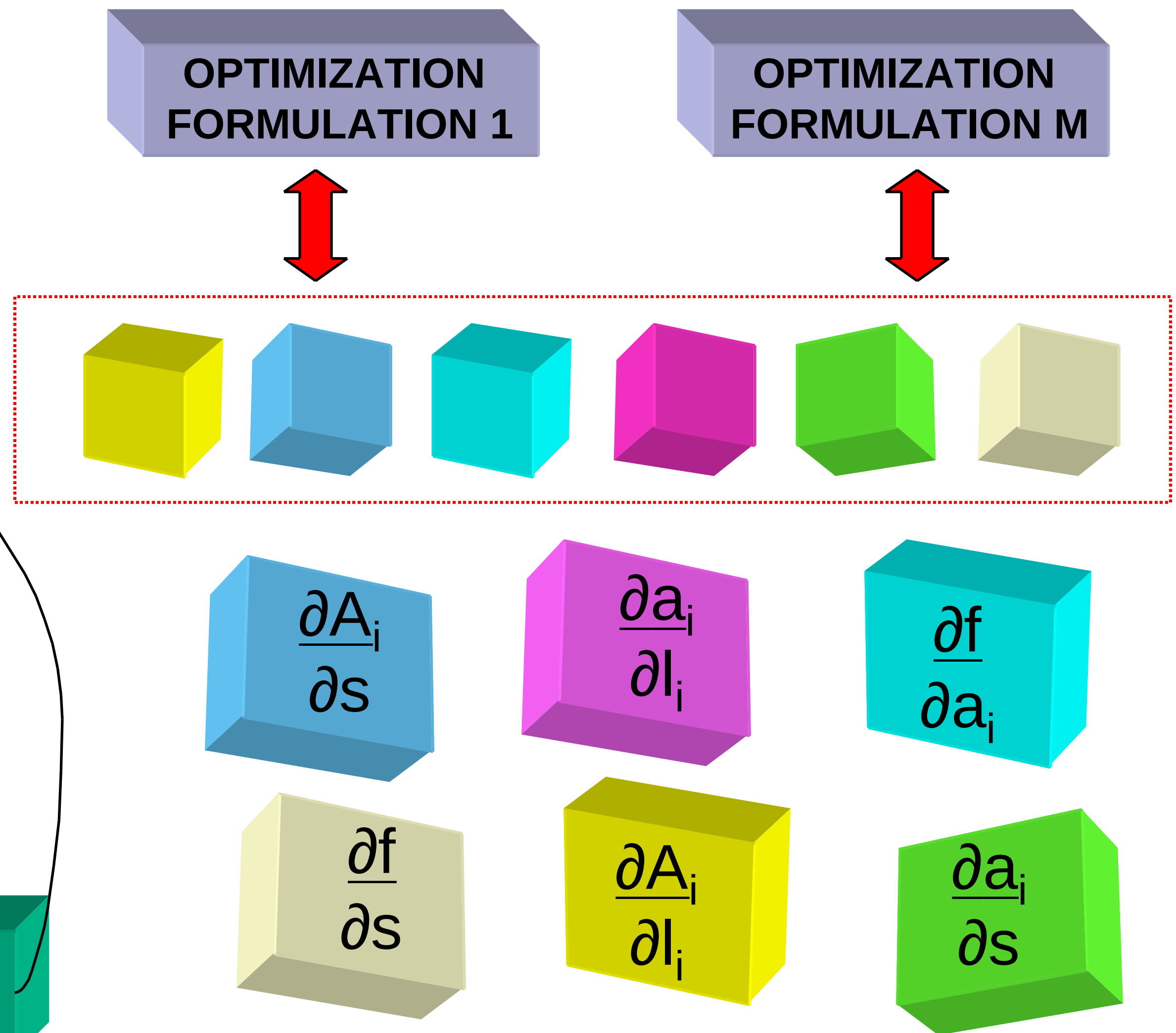
i = 1, …, N

NOW



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

FUTURE

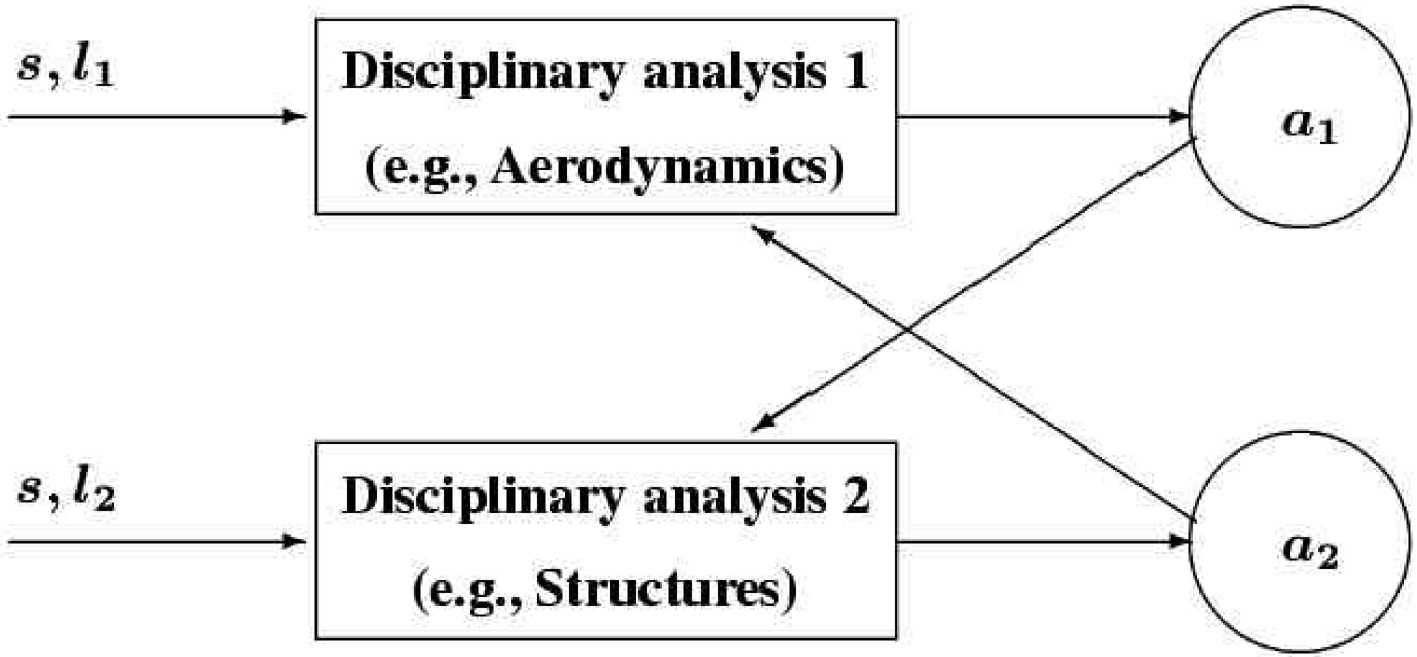


Expend the effort at the outset to implement analysis and sensitivity modules; easy to transform and expand: an opportunity for a general framework

Some Algorithmic and Structural Considerations in Problem Formulations

- Amenable to solution?
- Robust formulation?
  - Does the structure respect the canonical problem structure?
  - Do answers satisfy necessary conditions?
  - Are answers sensitive to small changes in parameters?
- Efficiency of solution?
- Autonomy of implementation / ease of transformation?
  - Claim: This is the most labor intensive part
- Autonomy of execution?
  - Wish to follow organizational structure for design
  - Wish to optimize only wrt local variables in disciplines

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**

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

Conventional Approach: Fully Integrated Formulation (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

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

Computational Properties of Distributed Formulations Based on Discrepancy Functions

- **"Discrepancy functions" – a device for distributing a coupled problem into autonomous subproblems**
  - Possibly do not perform multidisciplinary analysis at each iteration of the design optimization procedure
  - Quantify inconsistency among the shared variables and outputs of the disciplinary analyses as a scalar, whose value is obtained in disciplinary optimization subproblems
  - Drive the discrepancy function to zero at the solution of the problem (i.e., multidisciplinary analysis satisfied at the solution)
  - Manipulate local disciplinary variables in disciplinary optimization subproblems

Computational Properties of Distributed Formulations

- **Distributed optimization**
  - Representatives: Collaborative Optimization, Optimization by Linear Decomposition, Hybrids
  - Local variables handled in subsystems
  - Bilevel optimization
  - Unavoidable breakdown of the necessary optimality conditions or non-smoothness of the constraints
  - Difficult and expensive to solve by existing optimization methods
- **An alternative: distributed analysis opt (DAO)**
  - Single-level optimization
  - Autonomy of implementation available
  - Local variables treated at system level optimization
  - Robust wrt optimization algorithms
  - Autonomy of implementation similar to distributed opt

(Publications on LaRC LTRS)

Observations

- All desirable properties of MDO formulations (efficiency, robustness, subsystem autonomy of execution, etc.) are likely not be attainable in a single formulation
- All formulations are related to each other and require roughly the same amount of work to implement
- Computational elements needed for optimization can be implemented autonomously by disciplines
- Can reconfigure the same set of computational elements to implement one discipline or another
- Enable hybrid approaches

Example: Computational Elements in FIO and DAO

Recall 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: Computational Elements in FIO and DAO, 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: Computational Elements in FIO and DAO, cont.

Consider the Distributed Analysis Optimization approach (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: Computational Elements in FIO and DAO, 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

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

Algorithmic Interactions

- The same elements needed for FIO and DAO (and all distributed MDO formulations)
- Can implement elements with disciplinary autonomy if do not integrate MDA via fixed-point iteration early
- Elements integrated differently in FIO and DAO
- Can re-arrange computational components associated with one formulation and obtain components for another (may require substantial effort)
- For some formulations
  - Minor changes in an optimization algorithm yield an algorithm for solving another formulation
  - Straightforward to pass among some formulations $\Rightarrow$enable hybrid approaches: may use one far from solution, another near solution

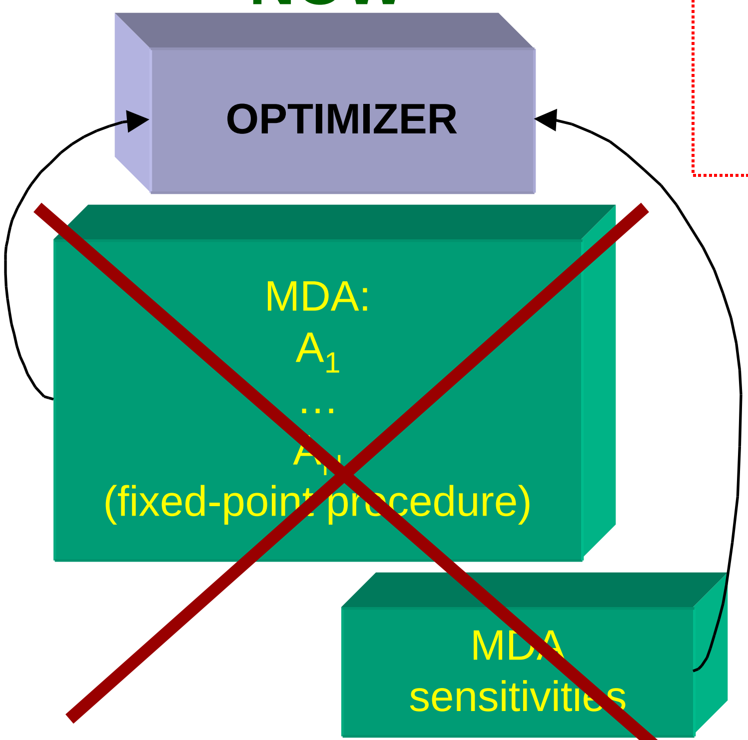
(Details can be found in publications)

Summary: MDO Synthesis: One-of-a-Kind vs. Modular, Reusable

Problem: design for objective f with

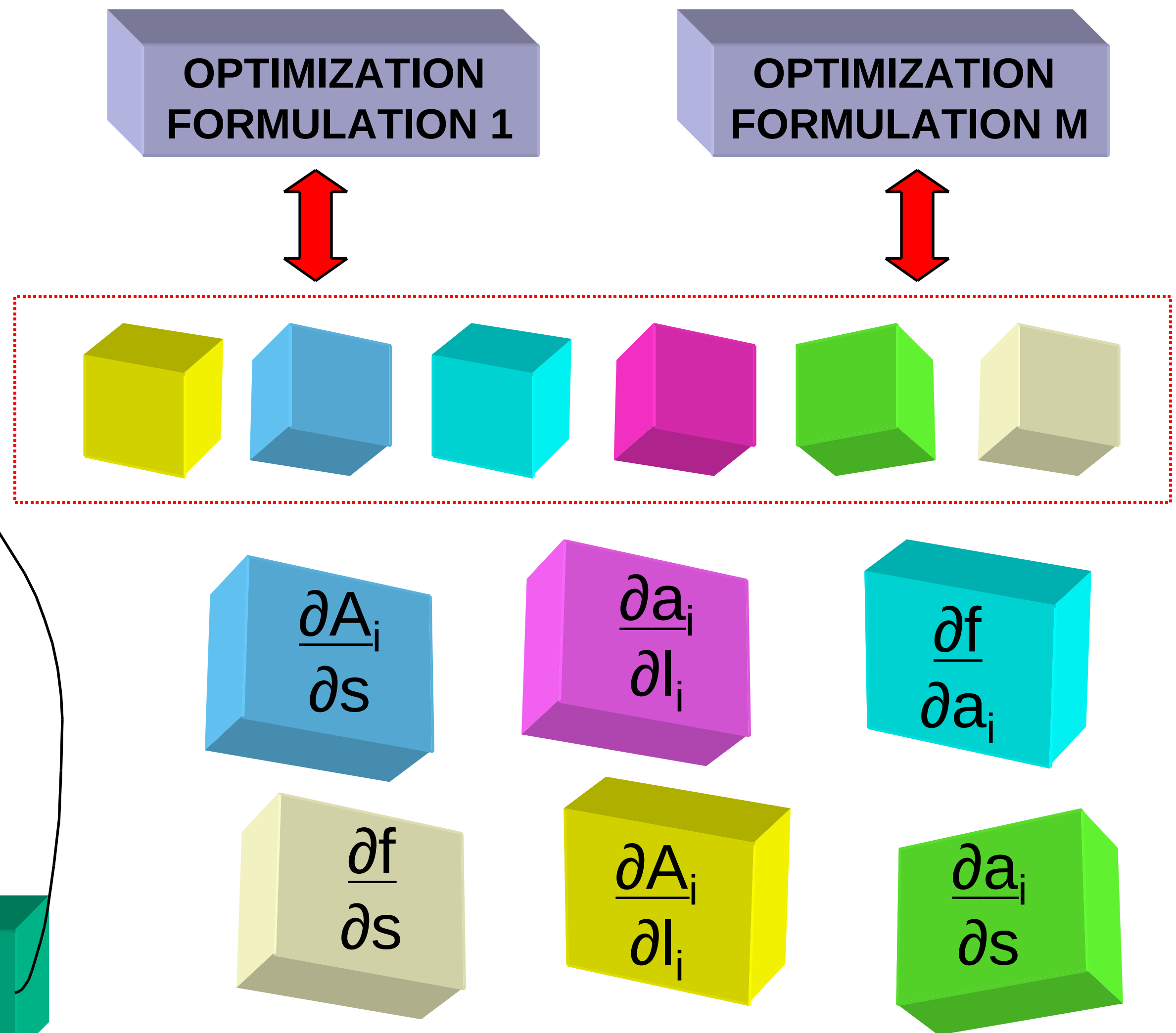
i = 1, …, N

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

Major Open Issues

- MDA for strongly coupled problems
- Variable-fidelity model management for MDO synthesis
- The most appropriate approach to multiobjective optimization
- A reliable distributed optimization approach
- A large-scale demonstration problem

FY01 Plans

- **Participants**
  - LaRC/MDOB, ODU, William & Mary
- **Process-based cost models**
  - Develop cost models for launch vehicle wing and fuselage
- **MDO problem formulations**
  - Complete analysis of modular MDO synthesis techniques in the context of FIO and one distributed problem formulation
- **Multiobjective optimization algorithms**
  - Develop or select a preliminary suite of multiobjective MDO tools
- **Demonstration**
  - Demonstrate multiobjective optimization on a medium-sized cost-performance launch vehicle MDO problem