Managing Models in Simulation-Based Design

Natalia M. Alexandrov
NASA Langley Research Center
Hampton, Virginia

Outline:

- Problem setting
- Model management as means to decrease uncertainty and cost
- Other uncertainty issues
- If time, software development environment

Fields Institute, Toronto, November 2005

Setting

- Design of engineering systems, characterized by design variable vector x
- Forward or analysis problem
  - Specify x; solve the governing partial differential equation (or equations) for intermediate field or state variable u
  - Evaluate outputs of engineering interest, f, based on u
- Design problem
  - Formulate design objectives and constraints based on f
  - Find the best value of f

Requirement for models

- Conflict of model uncertainty (a large impediment to practical acceptance of PDE-based design) and cost
  - Must be sufficiently fine (high-fidelity) so that the outputs and their derivatives represent system performance
  - Must be sufficiently coarse to be affordable for repeated use within design context
- Some approaches to resolving the conflict
  - Reduce/manage uncertainty associated with analysis and design
  - Use higher-fidelity models at earlier stages of design
  - Improve tractability of high-fidelity models in all stages of design

Focus: CFD-based aerodynamic optimization

- Progression of state of the art in CFD
  - Structured grids
  - Heuristic solution adaptation
  - Overset and unstructured grids
  - In progress
    - Adaptive analysis with error bounds/estimates
    - Adaptive design optimization with well-defined error bounds
  - Future
    - Multiscale problems

Environment – the FUN3D Suite

- Direct development team of about 10 people works on analysis and design using the RANS equations on 3D unstructured grids
- Discrete adjoint formulation used as a basis for error estimation, grid adaptation, and design
- Elasticity PDE formulation used for moving mesh applications
- Automated complex-variable conversion is used for direct differentiation
- A unique capability: exact dual integration algorithm used for computing hand-coded discrete adjoint for full RANS discretization

Compute derivatives via adjoints

Combine cost function f with Lagrange multipliers L_f and L_g to form Lagrangian, L:

$$L(\mathbf{D},\mathbf{Q},\mathbf{X},?_f,?_g) = \underbrace{f(\mathbf{D},\mathbf{Q},\mathbf{X})}_{\text{Objective: Lift, drag, boom, etc.}} + ?_f^T \underbrace{\mathbf{R}(\mathbf{D},\mathbf{Q},\mathbf{X})}_{\text{Flow Equations}} + ?_g^T \underbrace{(\mathbf{K}\mathbf{X} - \mathbf{X}_{surface})}_{\text{Mesh Equations}}$$

Differentiate with respect to **D**:

$$\frac{dL}{d\mathbf{D}} = \frac{\partial f}{\partial \mathbf{D}} + \left[\frac{\partial \mathbf{R}}{\partial \mathbf{D}}\right]^T ?_f + \left[\frac{\partial \mathbf{Q}}{\partial \mathbf{D}}\right]^T \left\{\frac{\partial f}{\partial \mathbf{Q}} + \left[\frac{\partial \mathbf{R}}{\partial \mathbf{Q}}\right]^T ?_f\right\}$$
$$+ \left[\frac{\partial \mathbf{X}}{\partial \mathbf{D}}\right]^T \left\{\frac{\partial f}{\partial \mathbf{X}} + \left[\frac{\partial \mathbf{R}}{\partial \mathbf{X}}\right]^T ?_f + ?_g^T \mathbf{K}\right\} - ?_g^T \left[\frac{\partial \mathbf{X}}{\partial \mathbf{D}}\right]_{surface}$$

$$\left[\frac{\partial \mathbf{R}}{\partial \mathbf{Q}}\right]^T ?_f = -\frac{\partial f}{\partial \mathbf{Q}}$$

Flowfield Adjoint Equation

$$\mathbf{K}^T ?_g = -\left\{\frac{\partial f}{\partial \mathbf{X}} + \left[\frac{\partial \mathbf{R}}{\partial \mathbf{X}}\right]^T ?_f\right\}$$

Mesh Adjoint Equation

$$\frac{dL}{d\mathbf{D}} = \frac{\partial f}{\partial \mathbf{D}} + ?_f^T \frac{\partial \mathbf{R}}{\partial \mathbf{D}} - ?_g^T \left[\frac{\partial \mathbf{X}}{\partial \mathbf{D}}\right]_{surface}$$

Sensitivity Equation

(Courtesy Eric Nielsen)

Benefits of adjoints

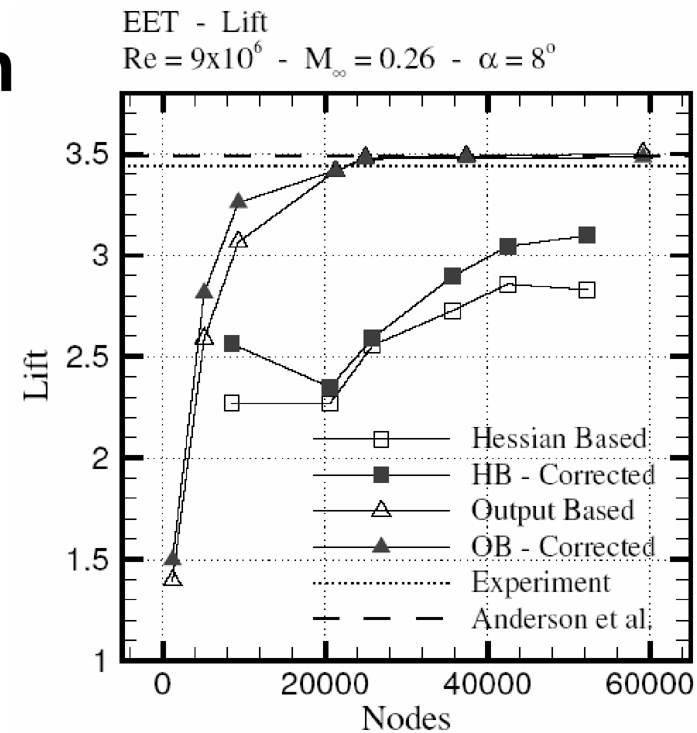
- Adjoints relate the local equation error to outputs of engineering interest providing
  - Rigorous error estimation and grid adaptation metric
    - No need for *a priori* knowledge of the flowfield
    - No reliance on heuristics for adaptation
    - Natural stopping criterion for convergence
    - Increases efficiency, reduces cost and uncertainty in analysis and design
  - The most efficient means of computing derivatives for functions of high dimensionality
    - Unlike other methods (e.g., finite differencing), independent on dimensionality

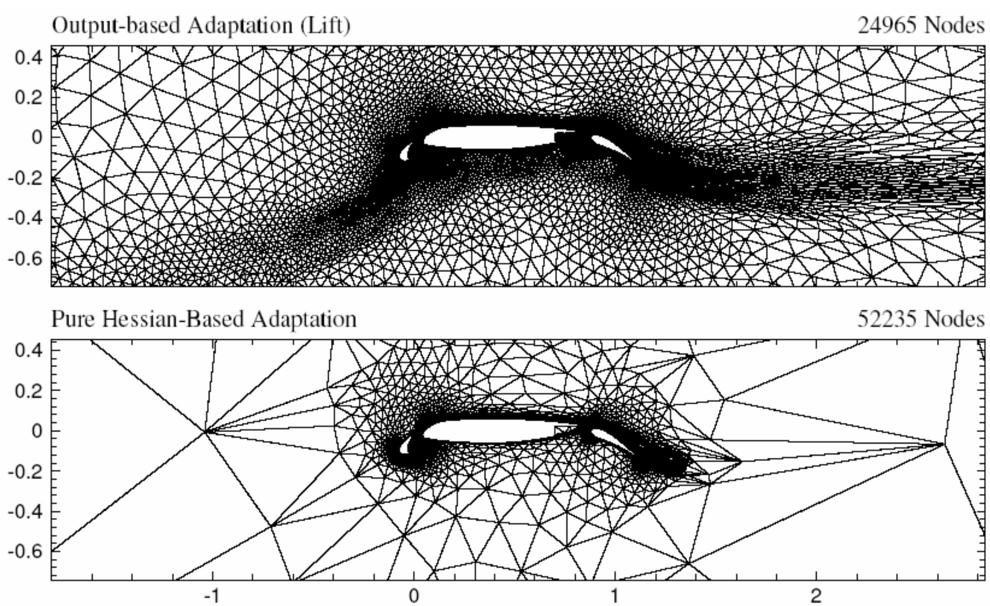
Adjoint-Based Mesh Adaptation

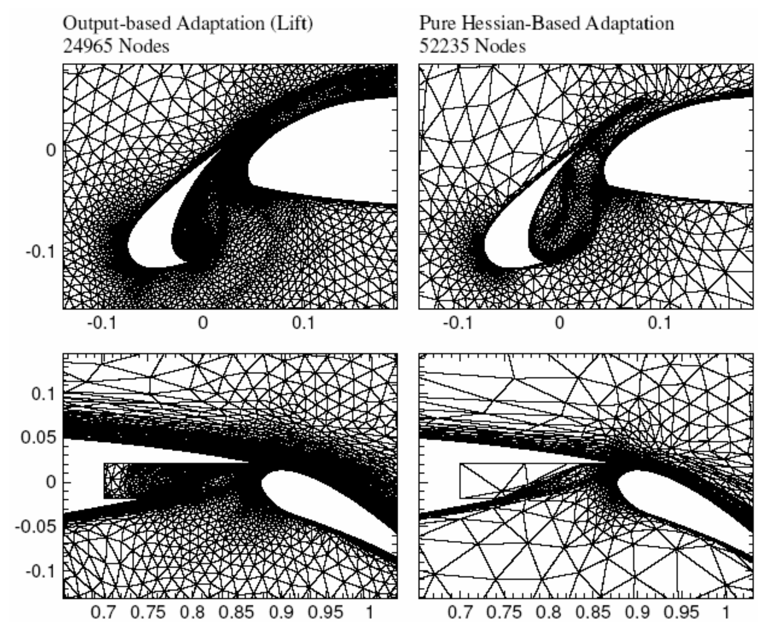
2D High-Lift Example

- Need accurate lift evaluation
- The initial mesh is coarse
- The adjoint-based technique recognizes important regions of the flow

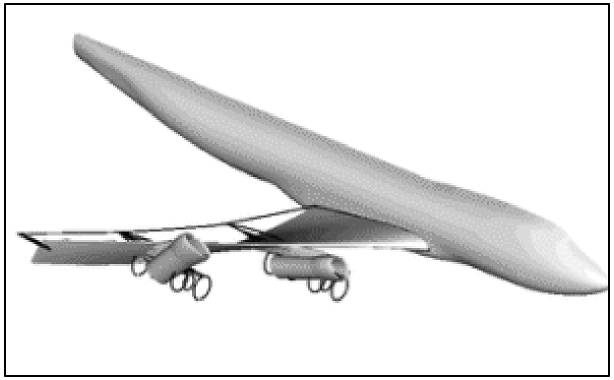
Courtesy Venditti & Darmofal at MIT (using FUN2D)



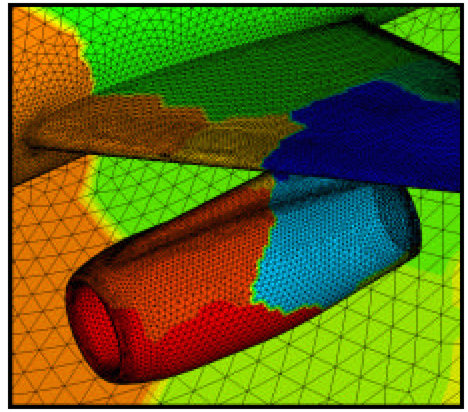




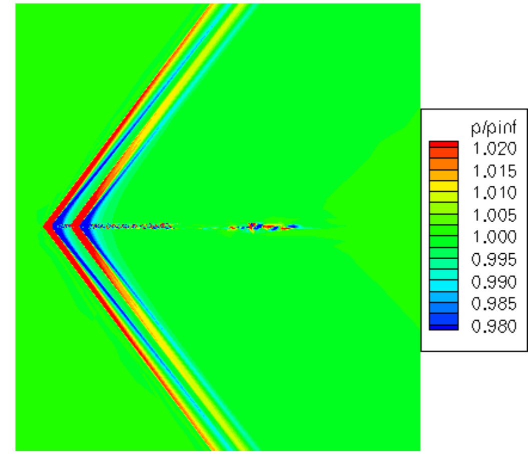
The FUN3D design environment



Parameterization

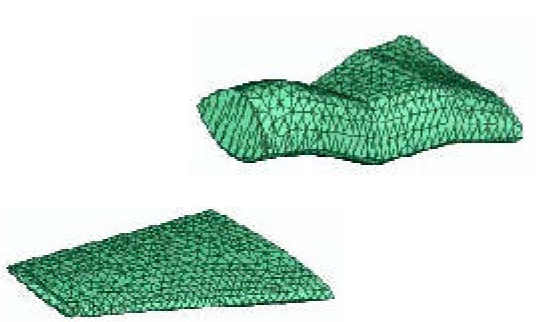


Domain Decomposition



Flow Solver: *f* (e.g. C_D)

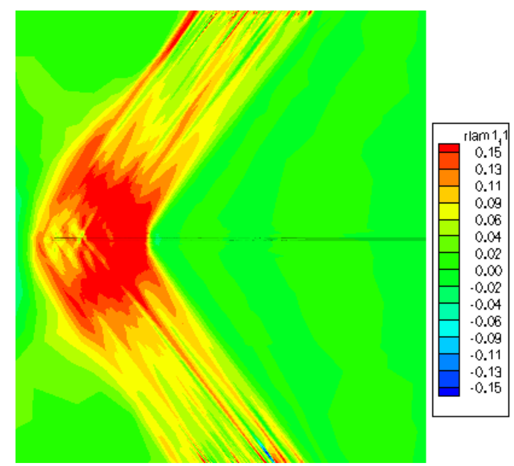
Minimize(*f*)



Mesh Movement

$$\nabla f$$

Derivative Evaluation



Adjoint Solver: ? $_f$

Specifics of simulation-based optimization setting

- Despite quality and efficiency of models, expensive and not very robust function & derivatives
  - Pervasive efforts in improving tractability of hi-fi models in design optimization (Sandia, Boeing, INSEAN, to name a few)
- Assume a set of models (hi/lo-fi) of varying accuracy/cost, with no information about model relationship or structure
  - E.g., variable-resolution, varying convergence, variable-fidelity physics, etc.
  - Special model structure or context provides additional algorithmic possibilities (e.g., variable-resolution models in multigrid context, Lewis and Nash; Gratton, Toint, Sartenaer)
- "Preliminary" design / hi-fi models: min. $O(10^2\text{-}10^3)$ variables
  - Must rely on derivative-based optimization
  - In lower-dimensional problems, model management can rely on derivative-free optimization and data sampling models (e.g., Booker et al., numerous other efforts); Space Mapping (Bandler et al.) is another approach; similarity of trends in models important in 0-order approximations
- Assume black-box function evaluation
  - Many efforts in problem formulation
  - Simultaneous analysis and design methods not current focus, but discussion on models applies

Addressing tractability of high-fidelity models

- 1st order Approximation and Model Management Optimization (AMMO) (e.g., NMA & Lewis, AIAA-96-4101/02)
  - Replace local Taylor-series models in subproblems of NLP algorithms with available lower-fidelity models (heuristic use of lo-fi models long-standing in engineering)
  - No reason to assume that lo-fi model trends follow those of hi-fi model $\Rightarrow$ impose local consistency conditions, i.e., assure local similarity of trends
  - AMMO can be imposed on any algorithm; usually faster than the basic algorithm because lo-fi models have better global properties than local Taylor-series models

AMMO vs. single-fidelity model optimization

Single-fidelity trust-region algorithms

- Do until convergence
  1. At x_k build local models (Taylor series) of the objective and constraints based on information computed by hi-fi simulation
  2. Compute a trial step by solving a subproblem based on local hi-fi models
  3. Check improvement in hi-fi responses and update iterates
- End do

Variable-fidelity (AMMO) algorithms

- Do until convergence
  1. At x_k select a model from a suite of available lo-fi models and compute corrections based on hi-fi and lo-fi models so that 1^{st} order consistency holds
  2. Compute a trial step by solving a subproblem based on corrected lo-fi models, using standard techniques
  3. Check improvement in hi-fi responses and update iterates
- End do

AMMO: Convergence vs. Efficiency

- (Essentially) traditional trust-region convergence results apply
- Convergence analysis relies on enforcing local similarity of trends: if f_{HI} is a high-fidelity model and f_{LO} is a low-fidelity model, f^{corr} in optimization subproblem is required to be consistent to 1st order at each major iteration x_k:

$$f^{corr}(x_k) = f_{HI}(x_k) \quad \text{and} \quad \nabla f^{corr}(x_k) = \nabla f_{HI}(x_k)$$

- Exact consistency not needed, but easy to enforce
- Practical efficiency is problem/model dependent on
  - Global predictive properties of low-fidelity model
    - Data-fitting models – with sufficient sampling, good *global* predictive properties
    - Problem/model dependent for other models
  - Expense of low-fidelity model

Enforcing local consistency via corrections

- Additive: $f_{HI}(x) = f_{LO}(x) + a(x)$
- Multiplicative: $f_{HI}(x) = \beta(x)\, f_{LO}(x)$
- Approximating exact $a(x) = f_{HI}(x) - f_{LO}(x)$, $\beta(x) = f_{HI}(x) / f_{LO}(x)$ by linear (quadratic) Taylor series expansion about x_k guarantees 1st (2nd) order consistency. E.g., building

$$\beta_k(x) = \beta(x_k) + \nabla\, \beta(x_k)^T (x - x_k)$$

and setting $f_k^{corr}(x) = \beta_k(x)\, f_{LO}(x) \Rightarrow$ 1st order consistency at x_k (Haftka, 1991)

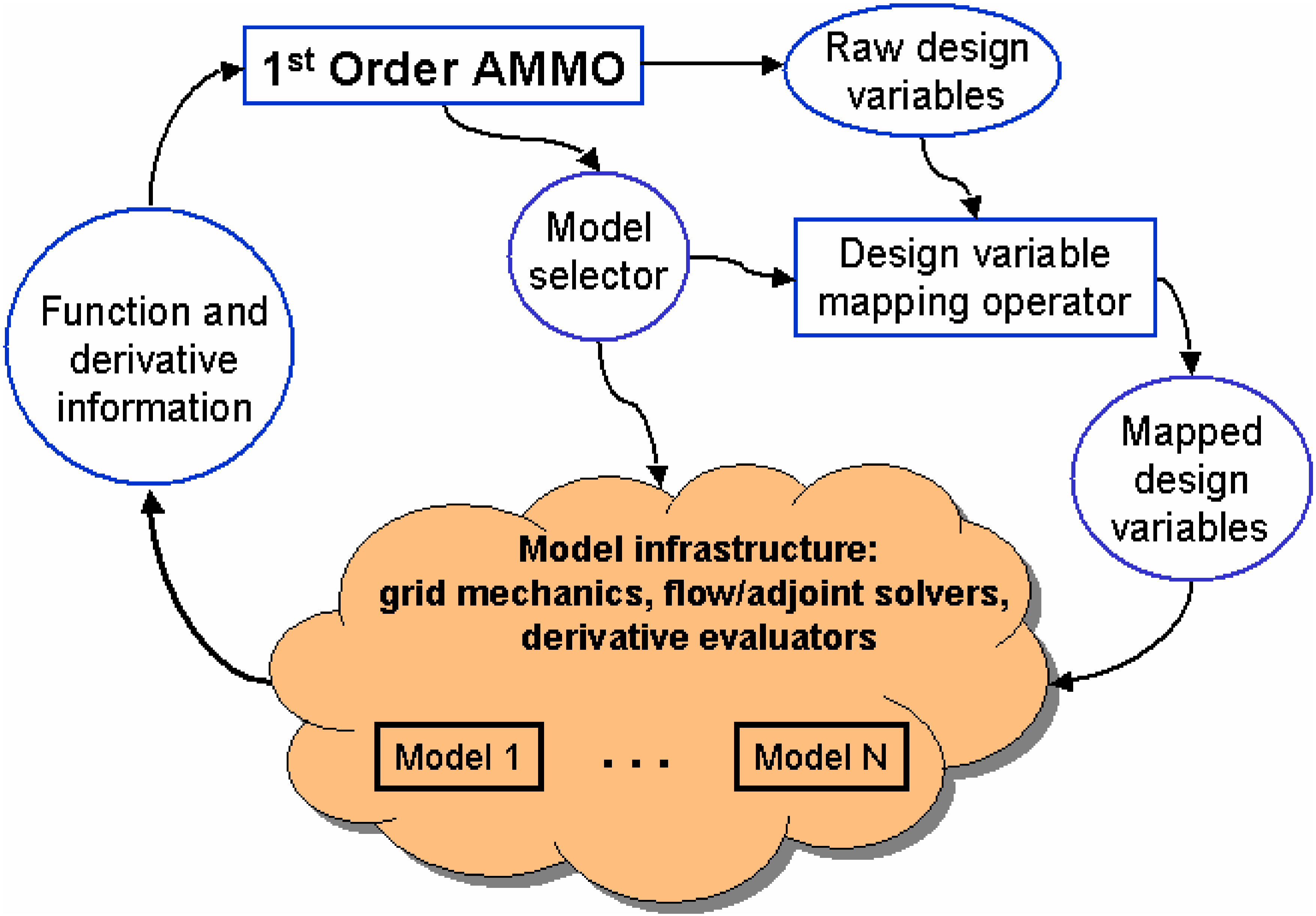
(Corrections can be mixed as necessary)

Computational experience

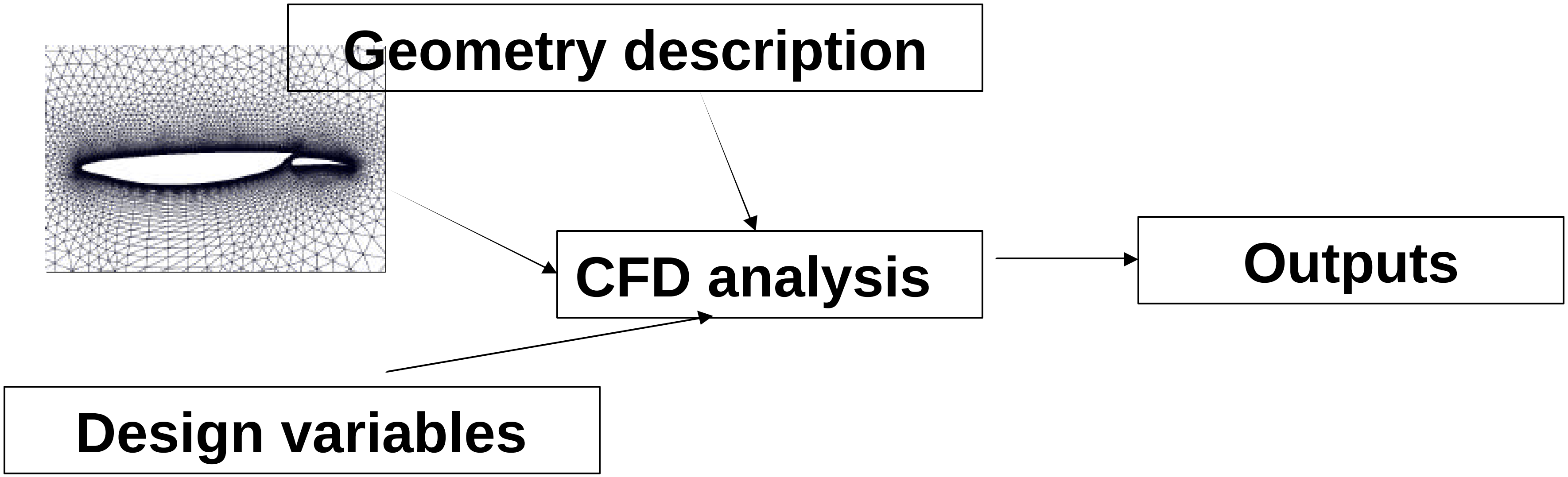
Several independent proofs of concept using AMMO for aerodynamic and hydrodynamic and MD design (e.g., Eldred, *et al*, Marduel *et al*., Campana & Peri)

- Typical savings in hi-fi function evaluations from 3 to 7-fold
- Q: If mimicking local Taylor series approximations, why expect any savings compared to conventional derivative-based methods?
- A: Local corrections, but hope that corrected lo-fi model has better global behavior. So far, has held for CFD-based applications.
- At NASA LaRC:
  - AMMO imposed on several algorithms (SQP, Augmented Lagrangian, a multilevel method)
  - Example: aerodynamic design optimization…

AMMO framework



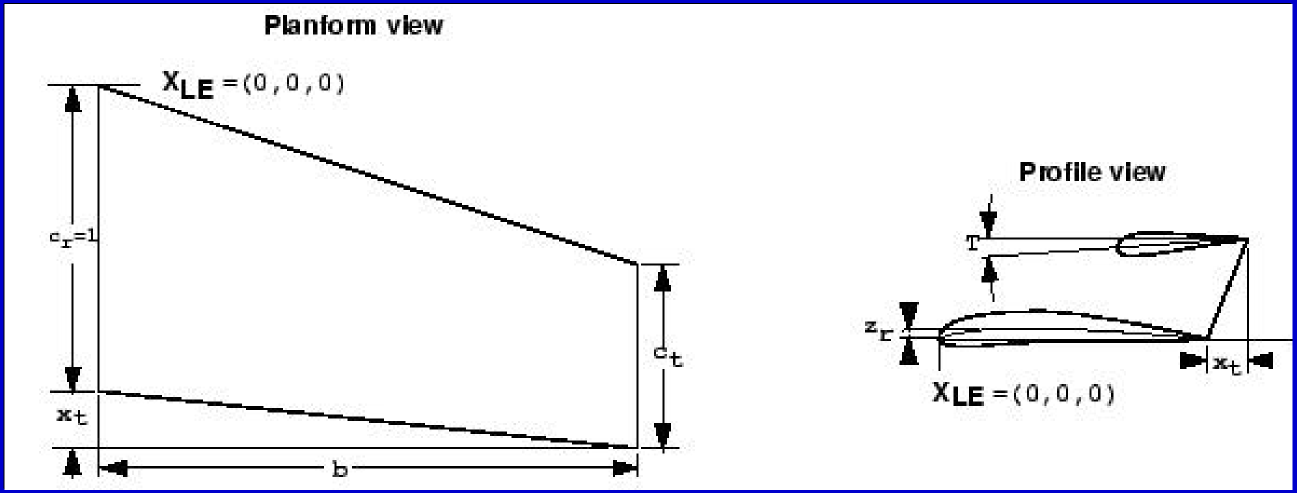
Example: aerodynamic shape optimization



- Minimize objectives (e.g., -L/D)
- subject to constraints on the moments

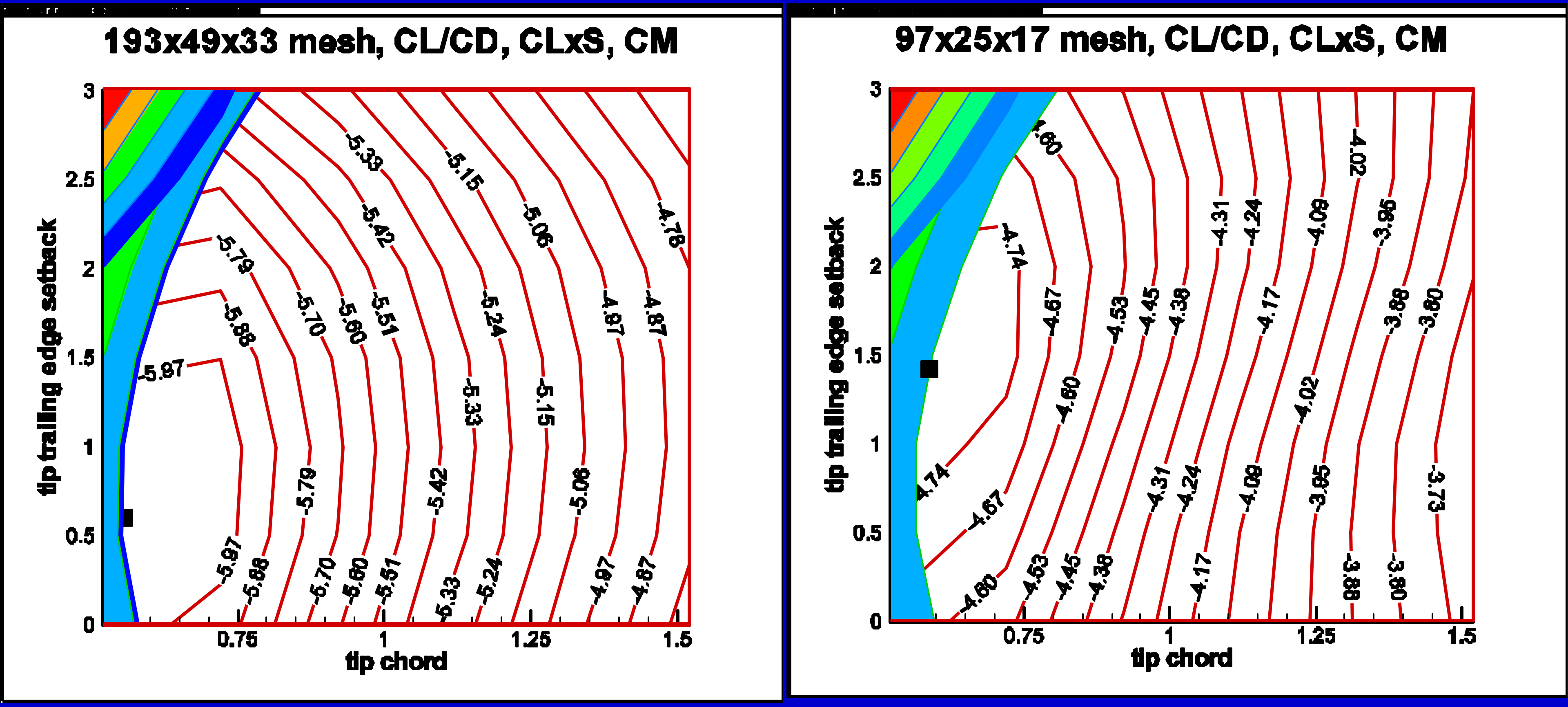
Summary of AMMO with variable resolution models:

- Wing design: minimize some combination of lift and drag, subject to constraints on the moments; same equations (Euler), varying grid refinement



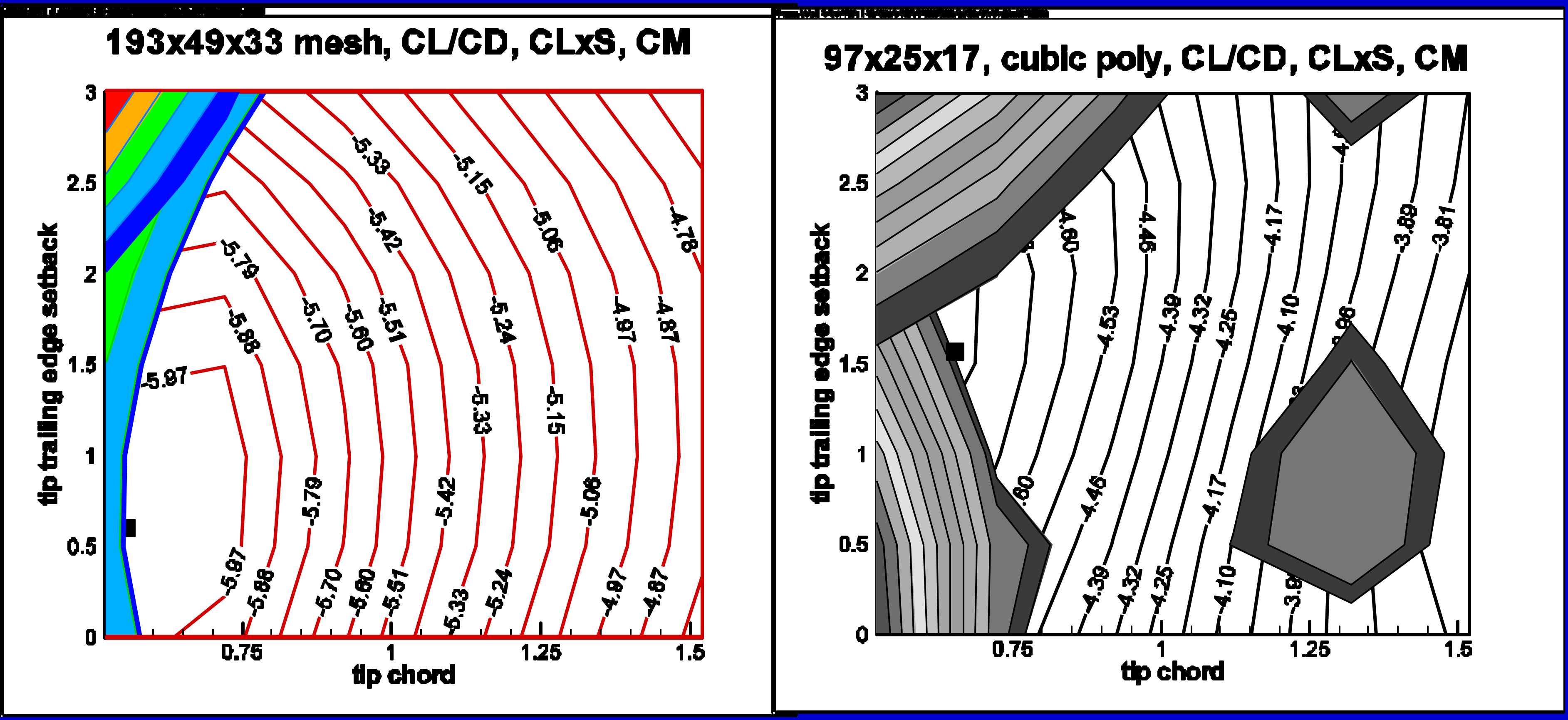
- Observations
  - If meshes were generated as proper subsets of one another, trends were similar
  - Functions computed on meshes that are not proper subsets of a mesh can result in large landscape variations

Favorable relationship between hi-fi & lo-fi model level sets



Trends in hi-fi model and even uncorrected lo-fi model are similar

Less favorable relationship between hi-fi & lo-fi model level sets

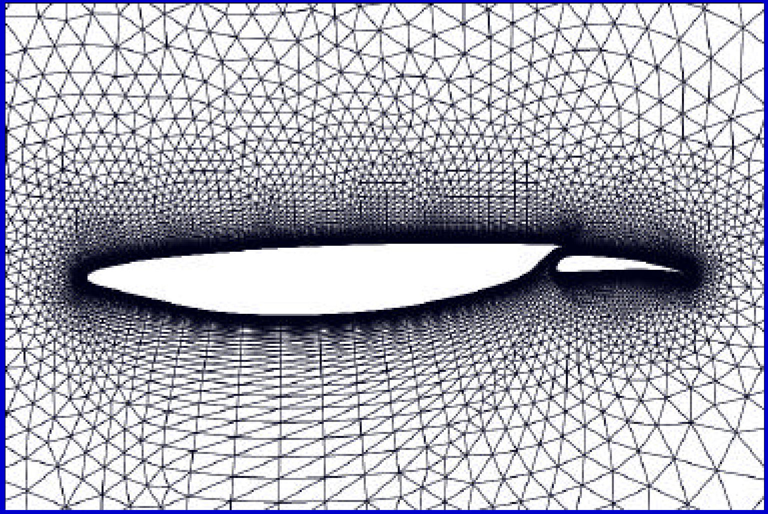


No impact on performance (subsonic, well-behaved problem); typical savings in terms of hi-fi evaluations 3-4 times (no tuning of algorithms)

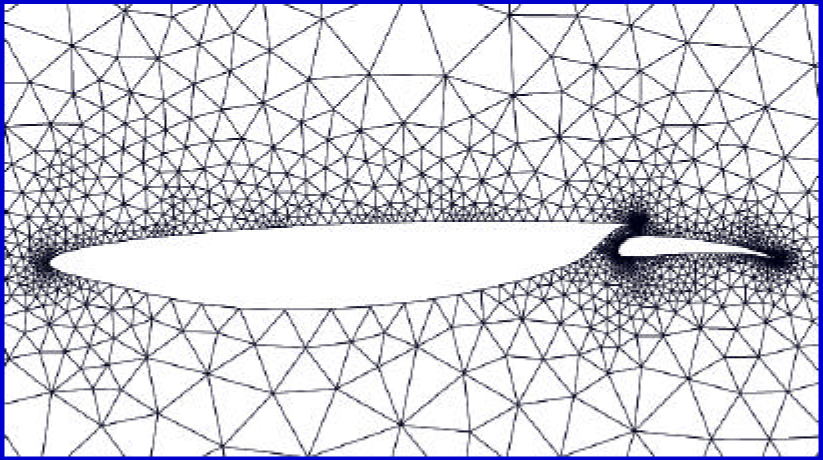
Summary of AMMO with variable resolution, variable-fidelity physics models

- Airfoil design; different equations (Navier-Stokes vs. Euler), varying grid refinement; posed as bound-constrained problem

Hi-fi grid

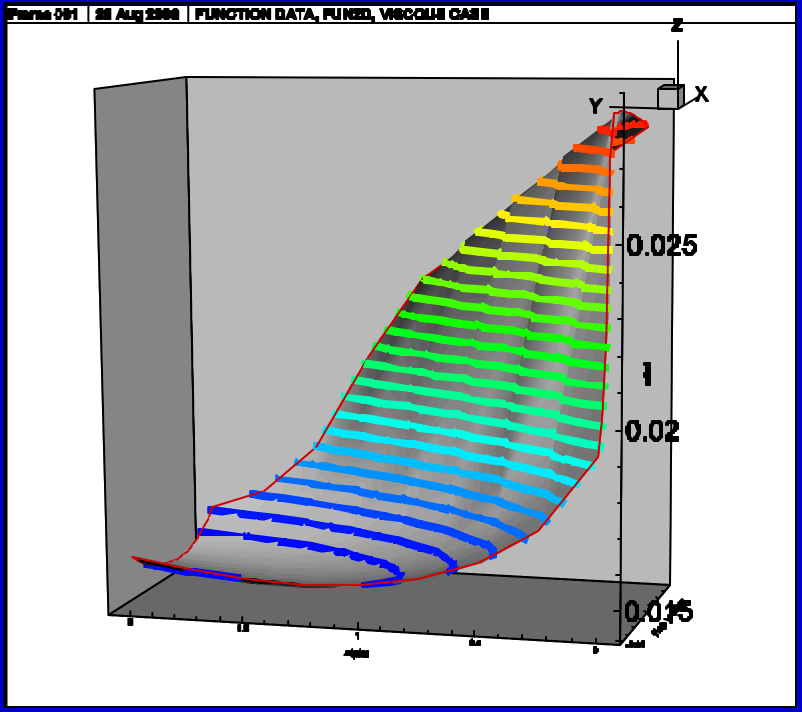


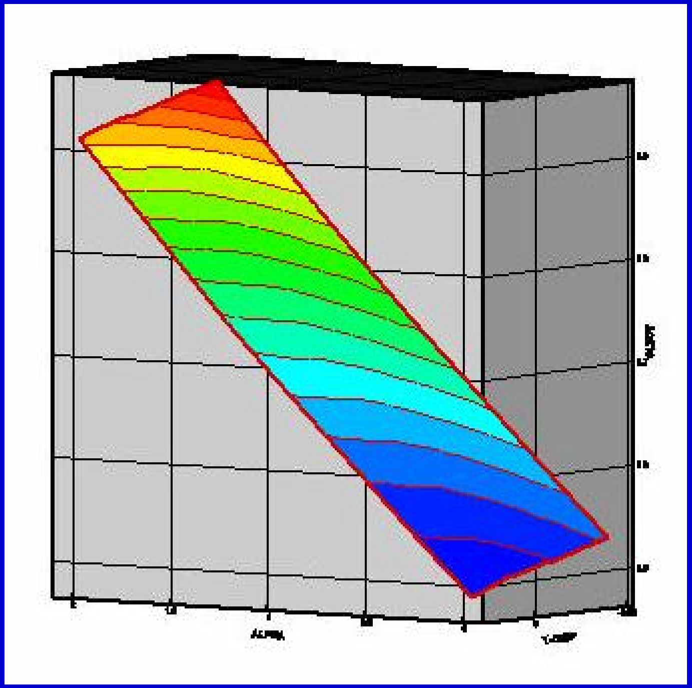
Lo-fi grid

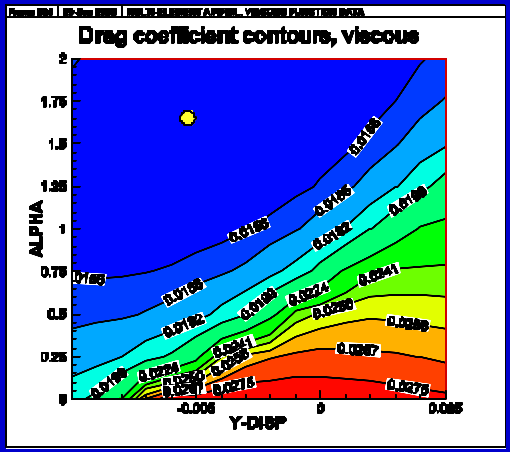


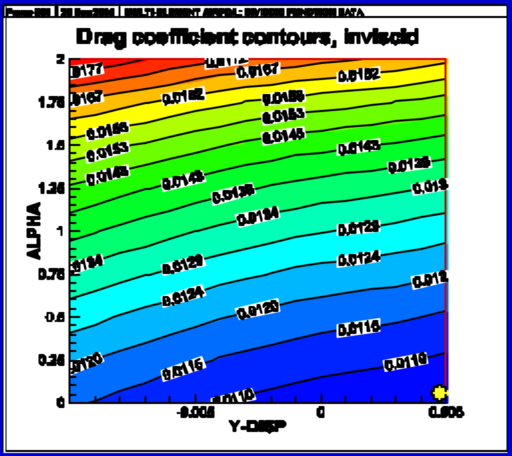
- Trends in models of different physical fidelity can differ drastically

Dissimilar models hi-fi and lo-fi models (combination of lift and drag)

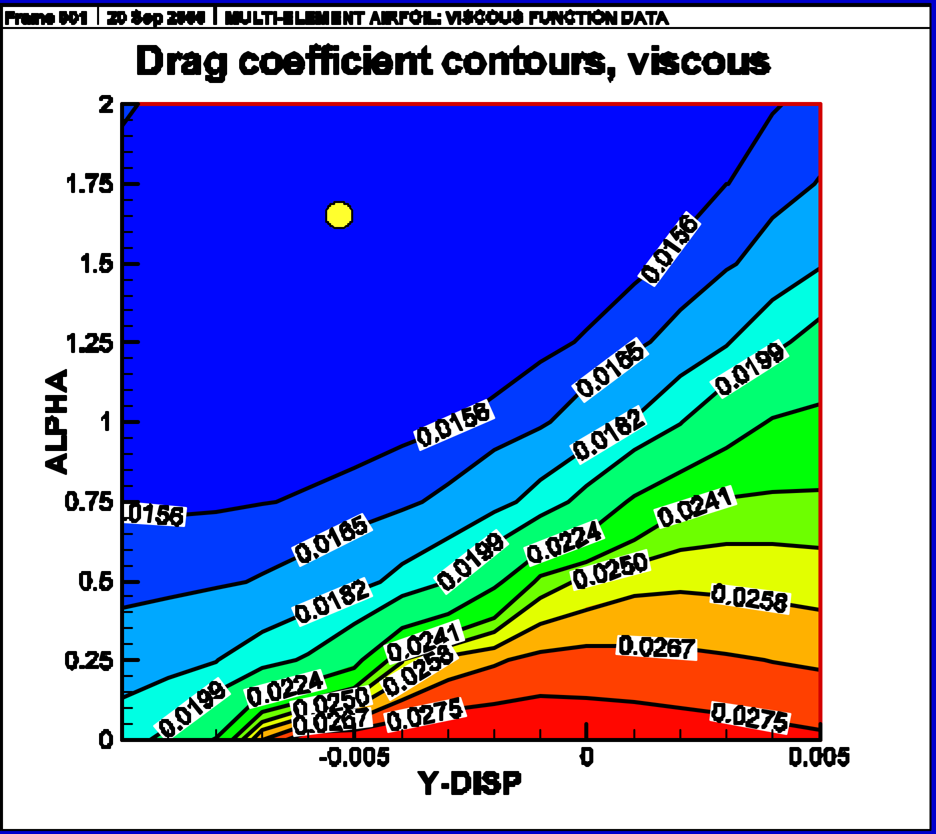


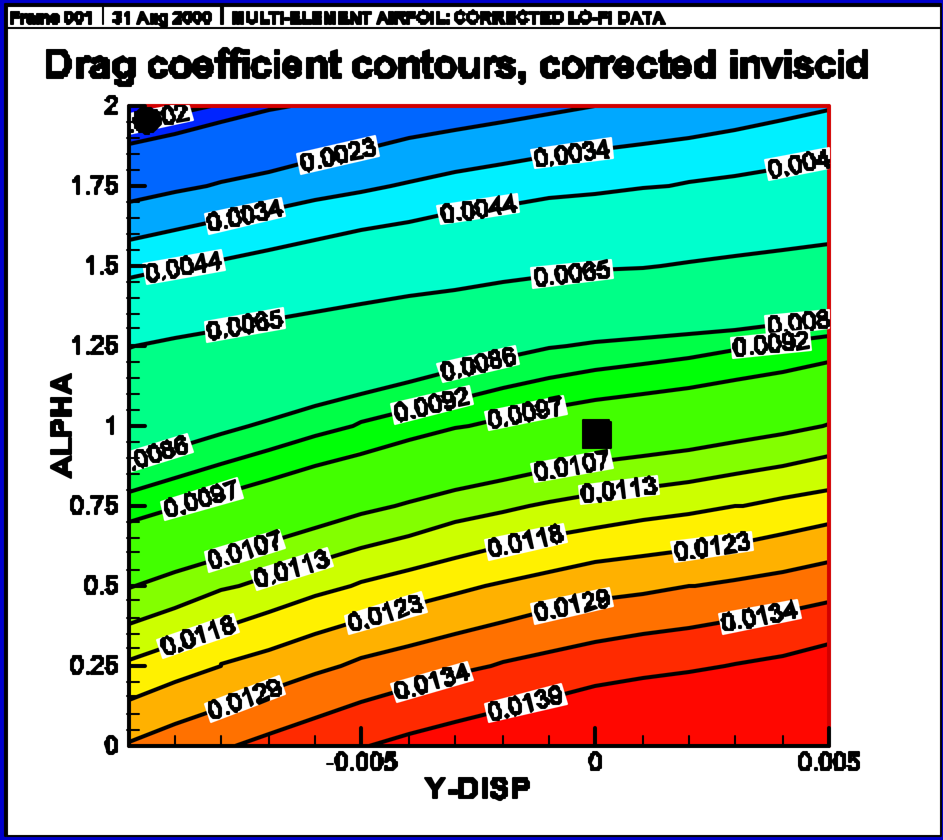






Hi-fi vs. corrected lo-fi model





First-order correction reversed trends

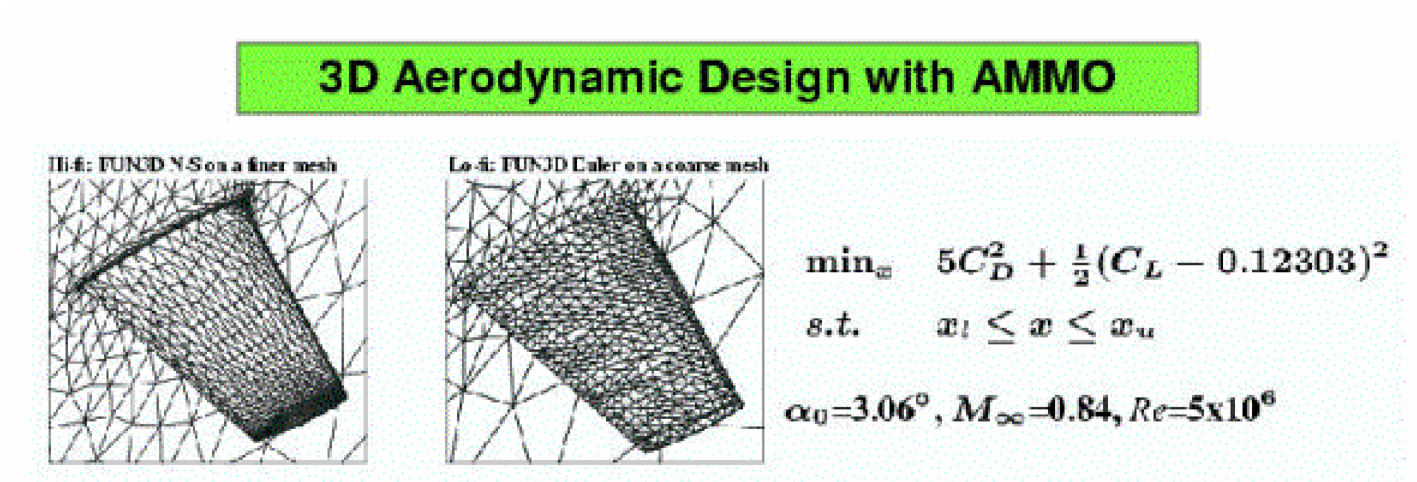
Efficiency depends on relative expense of low-fidelity model

Example: 2D (multi-element airfoil) aerodynamic optimization problem;
time/hi-fi analysis / time/lo-fi analysis » 120

	hi-fi eval	lo-fi eval	total CPU time	factor
Optimization (PORT), 2 variables	**14/13**		**» 12 hrs**	
AMMO, 2 variables	**3/3**	**19/9**	**» 2.41 hrs**	**» 5**
Optimization (PORT), 84 variables	**19/19**		**» 35 hrs**	
AMMO, 84 variables	**4/4**	**23/8**	**» 7.2 hrs**	**» 5**

(functions/gradients)

Savings depend on relative expense of low-fidelity model, cont.



- Approximately 4-fold savings in terms of hi-fi function evaluations
- Only 2 time savings; lo-fi model also expensive
- Developing “optimal” lo-fi models; distribute computations via PVD-like approach (after Ferris & Mangasarian)

Distributing computation (following Ferris & Mangasarian)

- Current problem: minimize $f(x)$, s.t. $x \in B$
- Notation: for $x \in R^n$, partitions are $x_l \in R^{nl}$, $l = 1, \ldots, p$ and nl sum up to n
- Let l* be a complement of l in {1, …, p}, $\mu_{l*} \in R^{p-1}$
- Let $d^k \in R^n$ be an arbitrary direction, partitioned into n subsets and

$$D^k_{l*} = \overset{p-1}{\begin{pmatrix} d^k_1 & & & & & \\ & \ldots & & & & \\ & & d^k_{l-1} & & & \\ & & & d^k_{l+1} & & \\ & & & & \ldots & \\ & & & & & d^k_p \end{pmatrix}} \, nl$$

- D^k_{l*} and μ_{l*} are used to form the “forget-me-not” term in subproblems

AMMO with PVD at a low-fidelity subproblem level

Given f_{HI}, x^0, and $B^0=B^{max}$ (bound constraints)

Do until convergence

1. Choose f^k_{LO} and compute correction s.t. $f^k_{corr}(x^k)=f_{HI}(x^k)$, $\nabla f^k_{corr}(x^k)=\nabla f_{HI}(x^k)$
2. Solve approximately for s^k: $\min_s f^k_{corr}(x^k + s)$ s.t. $x^k + s \in B^k$:

   Do until stopping criterion is satisfied{

   Solve in parallel: $\min_{x_l, \mu_{l*}} \varphi^k_l(x_l,\mu_{l*}) \equiv f^k_{corr}(x_l, x^k_{l*} + D^k_{l*}\mu_{l*})$

   s.t. $(x_l, x^k_{l*} + D^k_{l*}\mu_{l*}) \in B^k$,

   resulting in (y^k_l,μ^k_{l*})

   Synchronize: Compute x^{k+1} s.t. $f(x^{k+1}) \leq \min \varphi^k_l(y^k_l,\mu^k_{l*})$

   $s^k = x^{k+1} - x^k$
3. Update the iterate and bounds based on the actual decrease in f_{HI} produced by s^k vs. the decrease predicted by f^k_{corr}

}

Comments

- Forget-me-not term distinguishes PVD from block-Jacobi and coordinate descent (secondary variables are fixed there)

- Allowing secondary variables to move improves robustness of the algorithm, as observed in computations by Ferris and Mangasarian

- The choice of d^k is arbitrary theoretically, but important in practice; one particular choice (F&M) is scaled $-\nabla f(x^k)$ for unconstrained problems

- Following Solodov, we use the projected gradient residual function

$$d^k = r(x^k), \text{ where } r(x) = x - P_B\, [x - \nabla f(x^k)\,]$$

- Solodov's convergence theory allows for sufficient decrease instead of global solutions for the subproblems $\Rightarrow$ consequences for practical problems and parallelism

But point design is not enough

- One reason direct optimization is not used more widely for actual design is the lack of robustness in point optima
- Currently, much activity in uncertainty-based design; many approaches
  - Global uncertainty quantification
  - Incorporation into optimization
- In practice, in aerospace design, multipoint (robust) design = heuristic inverse design methods
  - Designers ignore “start with a dog” approaches
  - Design optimal airfoils for several flight conditions and average them
  - Average several target pressure distributions and design an airfoil to the resulting distribution
  - Etc.

Problem formulation

- Consider objective function $f : X \times Y \rightarrow \Re$

  $x \in X$ are design variables, controlled

  $y \in Y$ uncertainty variables, not controlled
- Ideally, find $x^* \in X$ s.t., $\forall\, y \in Y$

$$f(x^*;y) = f(x;y)\ \forall\, x \in X$$

- Example: aerodynamic shape design
  - Minimize drag of a *manufactured* airfoil; y are errors in manufacturing
  - Design an airfoil or a wing with good performance over a range of speeds and angles of attack; y are M_∞ and α

Problem formulation

- Central problem of statistical decision making
- Relax impossible problem…
  - via minimax principle (e.g., Ferguson 1967):
    - $\min_{x \in X} \psi(x) \equiv \sup_{y \in Y} f(x;y)$
    - conservative; protect against worst-case scenario
  - via Bayes principle (Welch et al. 1990):
    - $\min_{x \in X} \psi(x) \equiv \int_Y f(x;y)p(y)dy$
    - p is a probability density function on Y
    - minimize average loss; can be customized via p

Example: airfoil shape design

- $y = M_\infty$ and $f(x; M_\infty)$ is, say, the drag coefficient, then we want to solve

$$\min_{x \in X} \psi(x) \equiv \int_{\text{range of } M_\infty} f(x; M_\infty) p(M_\infty) dM_\infty$$

- Here p is the weight function that quantifies the value placed on performance at various speeds
- Such formulations studied by Huyse, Li, and others

Difficulty

- Tractability in question even for problems of medium size
- Under investigation (NMA and Trosset)
  - Low-fidelity models for integration (AMMO-like)
  - A variety of surrogates (data-fitting models) for integration

Now in the works, e.g.,

- Making use of special problem / model structure
  - Tighter integration of adjoint-based adaptation for CFD into optimization logic
  - Optimal modeling strategies, e.g., using lo-fi models of different geometric description (e.g., grid-based vs. non-grid based models); must deal with different variable domains
- PVD
  - General constraints
  - Multidisciplinary/multiobjective application