Optimization with Large-Scale CFD Simulations

Natalia Alexandrov
Multidisciplinary Optimization Branch
NASA Langley Research Center

http://mdob.larc.nasa.gov

11th SIAM Conference on Parallel Processing for Scientific Computing, 25-28 Feb 2004, San Francisco

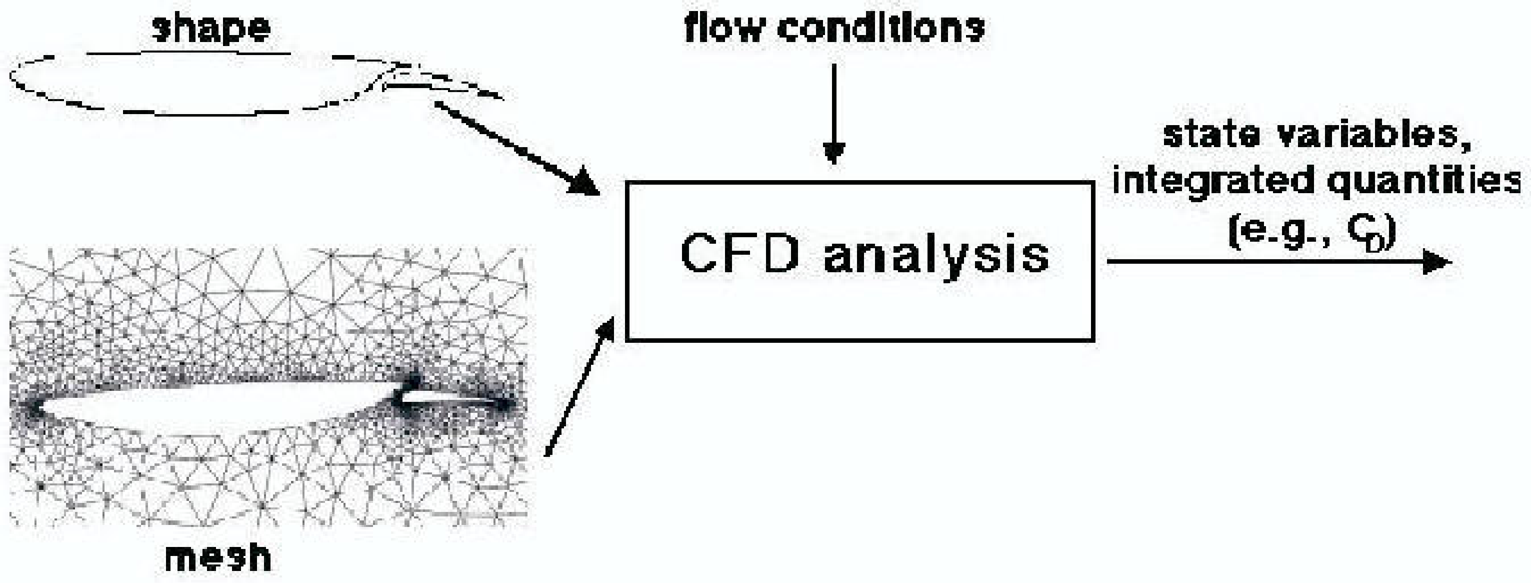
Optimization with simulations

- Problem

$$\begin{aligned} \min_{x} \quad & f(x, u(x)) \\ \text{s.t.} \quad & C_{E}(x, u(x)) = 0 \\ & C_{I}(x, u(x)) \leq 0 \\ & x_L \leq x \leq x_U \text{ with } u(x) \text{ solving } A(x, u(x)) = 0, \text{ given } x \end{aligned}$$

  - The use of parallelism depends on problem structure and formulation
    - At the level of function/derivative evaluations
    - Sequential optimization with parallel linear algebra
    - Parallel optimization algorithms
- Outline
  - Computational environment in Fast Adaptive AeroSpace Tools (FAAST) project
  - General difficulties in using CFD for function evaluations
  - A parallel optimization approach for single-discipline problems
  - Range of methods in the changing computational environment

Aerodynamic Optimization



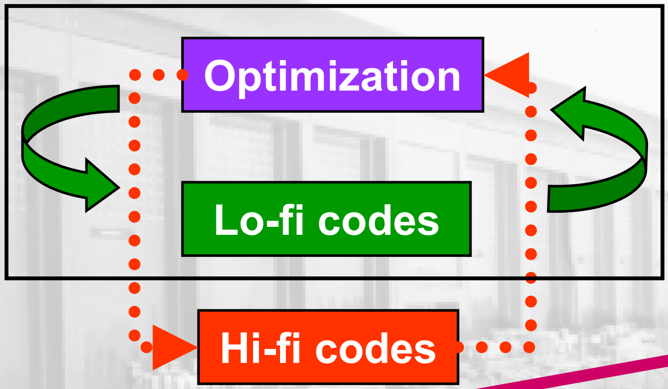
minimize Integrated quantities, such as $-\frac{L}{D}$ $(\frac{\text{lift}}{\text{drag}})$ or C_D (drag coefficient)

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

$x_l \le x \le x_u$

Fast Adaptive AeroSpace Tools (FAAST) Design Environment

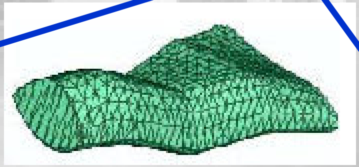
Objectives $f_i(x)$
Constraints $C(x)$
Derivatives $\nabla f_i(x)$, $\nabla C(x)$
for variable-fidelity models

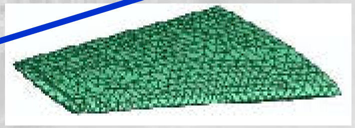


Design variables x
Based on choice of [illegible]

HARDWARE/
SOFTWARE
INFRA-
STRUCTURE

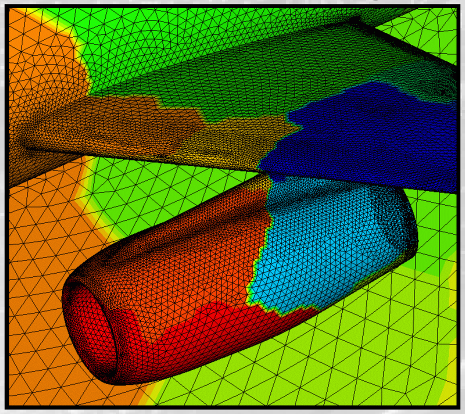
Choose or build model with input from mesh adaptation and error analysis





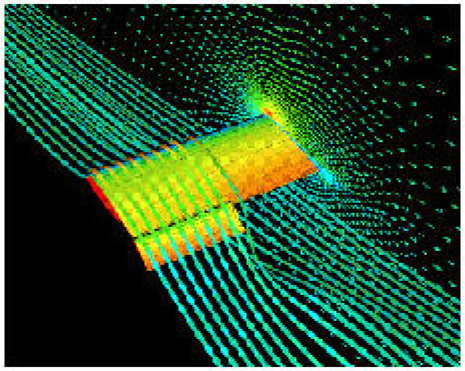
Mesh movement

Initialization:
Grid Generation
Grid Parameterization
Domain decomposition
for parallel processing



DESIGN
ORIENTED
ANALYSIS
SYSTEM

Mesh sensitivities
Derivative evaluation
Error analysis



Flow solver
Adjoint solver

General difficulties in CFD-based optimization

- Until recently, focused almost exclusively on preliminary design; among other problems, from the perspective of optimization:
  - Function evaluation is expensive and not sufficiently robust (e.g., unstructured mesh movement for viscous problems breaks often)
  - Function evaluation is not sufficiently automatic (e.g., grid regeneration)
- Recent additional emphasis on incorporating high-fidelity simulations into conceptual design
  - Conceptual design now uses simple, inexpensive models with an occasional recourse to a high-fidelity analysis
  - Design with high-fidelity simulations necessarily implies high dimensionality
  - CFD must be view in a multidisciplinary context (possibilities and challenges for parallelization)
  - Time is crucial in conceptual design

Current Computational Environment, cont.

- Function and derivative computations involve black-box CFD simulations $\Rightarrow$ Approach that now, arguably, yields the most efficient uses of parallelism—Simultaneous Analysis and Design (SAND)—is not realistic in the current environment of the project

- Number of variables and constraints not large in black-box formulation; “large-scale” refers to extreme expense of function evaluations $\Rightarrow$ No effort in parallelizing linear algebra in optimization algorithms

- Although most significant savings can be usually realized by addressing the special structure of an application, with the perspective of conceptual and multidisciplinary design, need general approaches, ability to combine the use of different models

- Treat PDE constraints implicitly and focus on optimization algorithms that make use of parallel function and derivative evaluation

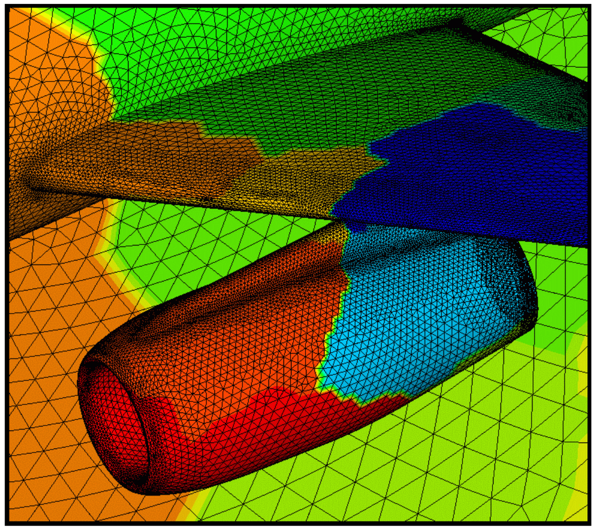
Flow Solver

FUN2D/3D
Fully Unstructured Navier-Stokes

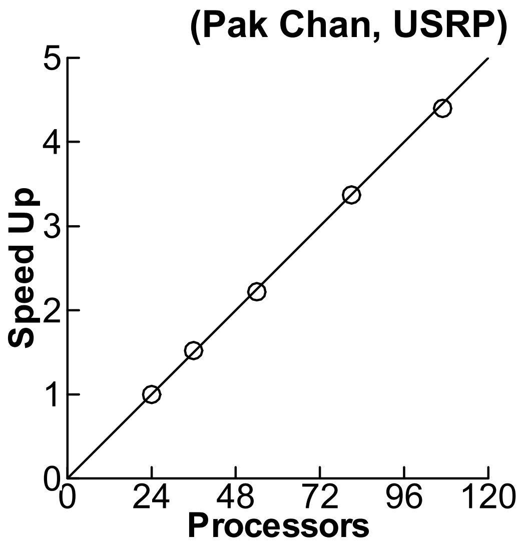
- FUN2D/3D (Anderson, Nielsen, FAAST) is a modular, state-of-the-art solver for turbulent flows on unstructured mixed-element grids across the speed range, with derivatives provided by adjoint approach
- Feasible to run only on multiprocessor systems
  - Expect 375 words of memory per grid point
  - Example: a grid with 1 million mesh points has about 6 million cells (tetrahedra) and 8.4 Gbytes
- Variety of solution algorithms available, including multigrid, Newton-Krylov, and others
- Supports a wide range of research projects and is in high demand for industrial and academic applications

Flow and Adjoint Solvers: Domain Decomposition and Parallelization

- Any sequential operations become showstoppers in terms of both memory and speed
- All components developed for distributed platforms
  - Parallel Party (D. Hammond) for grid partitioning (uses ParMETIS (http://www-users.cs.umn.edu/~karypis/parmetis/index.html)
  - Pre-/Post-processing: implicit line construction, multigrid interpolants, etc.
  - Mesh movement and adaptation (enrichment, coarsening, etc.)
- Flow solver parallelization scheme has shown excellent scaling on available hardware of over 100 distributed CPU's



Both strong and weak scalability studied.



Optimization Approach in FAAST

- Despite efficiency of flow and adjoint solver, straightforward optimization with high-fidelity simulations for large problems is prohibitively expensive

- Current approach: 1st order Approximation and Model management Optimization (AMMO) (e.g., Alexandrov & Lewis, AIAA-96-4101/02)
  - Combines the long-standing engineering practice of low-fidelity model use (heuristic, sometimes with corrections) with rigorous nonlinear programming techniques that guarantee convergence to high-fidelity answers
  - Available models: variable-accuracy, variable-resolution, variable-fidelity physics models; models based on sampling
  - Meant to make the use of high-fidelity models affordable in solving large-scale problems in conceptual and preliminary design

- Since 1996, several independent proofs of concept using AMMO for aerodynamic and multidisciplinary design; typical savings in hi-fi function evaluations from 3 to 7-fold

AMMO

Single-fidelity algorithms

- Do until convergence
  1. Build local models (usually 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 (globalization) and update iterates
- End do

Variable-fidelity (AMMO) algorithms

- Do until convergence
  1. Select a model from a suite of available lo-fi models and compute corrections based on hi-fi and lo-fi models so that 1st 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 (globalization) and update iterates
- End do

AMMO: Convergence vs. Efficiency

- Convergence relies on enforcing local similarity of trends: if f_{HI} is a high-fidelity model and f_{LO} is a low-fidelity model, f_{LO} is required to be consistent to 1st order at each major iteration:

  $$f_{LO}(x_k) = f_{HI}(x_k) \quad \text{and} \quad \nabla f_{LO}(x_k) = \nabla f_{HI}(x_k)$$

  Easily enforced for arbitrary pairs of functions via multiplicative or additive corrections, (e.g., Haftka, 1991)

- Practical efficiency is problem/model dependent on
  - Global predictive properties of low-fidelity model
  - Expense of low-fidelity model

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)

Efficiency depends on relative expense of low-fidelity model

Example: 3D (wing) aerodynamic optimization problem;
time/hi-fi analysis / time/lo-fi analysis ≈ 6

	hi-fi eval	lo-fi eval	total CPU time	factor
Optimization (PORT), 54 variables	13/11		≈ 175 hrs	
AMMO	3/3	22/15	≈ 86 hrs	≈ 2

(functions/gradients)

Efficient low-fidelity modeling aspects under investigation
- "Optimal" models
- "Optimal" termination of subproblems

Another approach: attempt to parallelize AMMO

Parallelization of AMMO

- Which basic algorithm to choose?
- In general, cannot exploit (partial) separability
- Investigate Parallel Variable Distribution (Ferris and Mangasarian, continued by Solodov)
  - Applicable to problems with convex constraints
- Distribute low-fidelity computations
- Cases for comparison
  - AMMO without PVD
  - AMMO with PVD in low-fidelity subproblem
  - PVD with single-fidelity (future)

Algorithm

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

$$D^k_{l^*} = \begin{array}{cc} p-1 & \\ \begin{pmatrix} d^k_1 & & & & & \\ & \dots & & & & \\ & & d^k_{l-1} & & & \\ & & & d^k_{l+1} & & \\ & & & & \dots & \\ & & & & & d^k_p \end{pmatrix} & nl \end{array}$$

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

AMMO with PVD at the 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 $(y^k_l,\mu^k_{l^*})$
                Synchronize: Compute $x^{k+1}$ s.t. $f(x^{k+1}) \le \min_{l \in \{1,\ldots p\}} \varphi^k_l(y^k_l,\mu^k_{l^*})$
            } $s^k = x^{k+1} - x^k$
    3. Update the iterate and bounds based on the actual decrease in $f_{HI}$
       produced by $s^k$ vs. the decrease predicted by $f^k_{corr}$
}
```

Some 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

Concluding remarks

- Fully coupled, expensive, black-box based problems are difficult to solve by parallel gradient-based optimization
- There are many approaches to parallel optimization
  - Zero-order methods (e.g., pattern search methods)
  - Multi-start derivative-based methods
  - In both derivative-based and derivative-free methods, construction of low-fidelity models from samples of high-fidelity model outputs (e.g., response surfaces, reduced order models, etc.)
  - Domain decomposition and explicit use of discretized analysis equations as constraints (SAND)
- There are few quantitative guidelines as to what approach may be beneficial under which circumstances
- Initiated a systematic comparison of a range of techniques
- Time will show…