A POD-Based Iterative Solver for Fast Structural Optimization

Kevin Carlberg and Charbel Farhat
Stanford University, USA
Department of Aeronautics & Astronautics

Seoul National University-Stanford University Student Joint Workshop 2009
Stanford, CA
June 18, 2009
Motivation

- Structural design optimization via computer simulation
  - Widely used in industry
  - Often very expensive due to repeated simulation

We propose an adaptive model
  - Accurately and inexpensively approximates performance
  - Leads to fast design optimization

Aero-structural optimization of ARW-2 wing
(courtesy Manuel Barcelos)
Motivation

- Design optimization:
  
  \[
  \begin{align*}
  \text{maximize} & \quad \text{Performance} \\
  \text{by changing} & \quad \text{Design variables} \\
  \text{subject to} & \quad \text{Constraints}
  \end{align*}
  \]

- Simulation-based design optimization:
  
  \[
  \begin{align*}
  \text{maximize} & \quad J(u, \mu) \\
  \text{by changing} & \quad \mu \\
  \text{subject to} & \quad c(u, \mu) = 0, \ d(u, \mu) \geq 0 \\
  & \quad K(\mu)u = f(\mu)
  \end{align*}
  \]

  \(K(\mu)\) stiffness matrix, \(u\) displacement, \(f(\mu)\) load vector

  \[\rightarrow\] High-fidelity finite element model of the structure.
Current Approach #1

- Embed high-fidelity model within a numerical optimization algorithm

Prohibitively expensive for large-scale structures
Surrogate-based optimization

- Current Approach #2
  - Optimize with a surrogate model
    1. Mimics the high-fidelity model
    2. Inexpensive to evaluate
  - Inaccurate away from calibration points
SBO Problem

- Trust-region model management (Alexandrov et al., 1998)
  1. Calibrate surrogate by high-fidelity simulation
  2. Optimize within a trust region using the surrogate
  3. Grow/shrink trust region depending on surrogate accuracy

- Global inaccuracy of surrogate leads to many costly calibrations

Goal: directly control approximation errors $\rightarrow$ lower overall cost
Use surrogate as acceleration tool for high-fidelity simulation

- Directly control the error (high-fidelity tolerance)
  - Global accuracy: fewer high-fidelity simulations
  - Improve accuracy as optimum is approached

![Graph showing high-fidelity response, surrogate response, and adaptive ROM response.](image)

- High-fidelity response
- Surrogate response
- Adaptive ROM response
- Calibration point
At each optimization iteration $k$, solve

1) State equations \[ K(\mu^{(k)}) u = f(\mu^{(k)}) \]

2) Sensitivity equations

- Direct S.A. For $i = 1, \ldots, n_{\text{vars}}$

\[ K(\mu^{(k)}) \frac{du}{d\mu_i} = \left. \frac{\partial f}{\partial \mu_i} \right|_{\mu^{(k)}} - \left. \frac{\partial K}{\partial \mu_i} \right|_{\mu^{(k)}} u \]

or

- Adjoint S.A. For $i = 1, \ldots, n_c + 1$

\[ K(\mu^{(k)}) \psi_i = \left. \frac{\partial \gamma_i}{\partial u} \right|_{\mu^{(k)}}^T \]

\[ \gamma_i = \begin{cases} c_i, & i = 1, \ldots, n_c \\ J, & i = n_c + 1 \end{cases} \]
Repeated analyses formulation

- For $k = 1, \ldots, K$ and $i = 1, \ldots, n_{\text{RHS}}$, solve
  \[ K(\mu^{(k)}) u_i = f_i(\mu^{(k)}) \]

  - $K(\mu^{(k)})$ large, sparse, symmetric positive definite (SPD)

- Iteratively solve by preconditioned conjugate gradient (PCG)
  - For $m = 1, \ldots, M$ (until convergence)
    \[
    \min_{x \in \mathcal{K}_m} \frac{1}{2} x^T K(\mu^{(k)}) x - x^T f_i(\mu^{(k)})
    \]
    - $\mathcal{K}_m$ Krylov subspace of dimension $m$
    - Final solution $\tilde{u}_i \in \mathcal{K}_M$ satisfies specified solver tolerance

- Approach: accelerate PCG convergence using ROM concepts
Adaptive ROM Approach

Solve \( K(\mu^{(k)})u_i = f_i(\mu^{(k)}) \) for \( k = 1, \ldots, K \), \( i = 1, \ldots, n_{\text{RHS}} \)

- Compute approximations \( \tilde{u}_i \) satisfying controlled tolerance \( \epsilon_k \)

\[
\frac{\| f_i(\mu^{(k)}) - K(\mu^{(k)})\tilde{u}_i \|_2}{\| f_i(\mu^{(k)}) \|_2} < \epsilon_k
\]

- Increase accuracy \( (\epsilon_k \to 0) \) as the optimum is approached

- Approximations lie in the sum of two subspaces

\[
\tilde{u}_i \in \mathcal{P} + \mathcal{K}_M
\]

- \( \mathcal{P} \) proper orthogonal decomposition (POD) subspace

- Compute \( \tilde{u}_i \) very efficiently by a novel augmented conjugate gradient (CG) iterative method
 Proper Orthogonal Decomposition

- Optimal representation of “snapshot” data

Original: rank 200  POD rank 50 approximation  POD rank 10 approximation
Optimal representation of “snapshot” data

Here, approximately minimize the projection error of the solution at a target configuration \( \bar{\mu} \) near \( \mu^{(k)} \)

1. Snapshots \( \{ w_j \}_{j=1}^{n_w} \): components of solution \( u(\bar{\mu}) \)
   - Solution at previous configurations
   - Sensitivity derivatives (Carlberg and Farhat, 2008)

2. Weights \( \{ \gamma_j \}_{j=1}^{n_w} \): estimate the solution
   
   \[
   u(\bar{\mu}) \approx u_{\text{est}}(\bar{\mu}) = \sum_{j=1}^{n_w} \gamma_j w_j
   \]
   - Radial basis functions & Taylor expansion coefficients

3. POD norm: \( \| x \|_{K(\bar{\mu})} \equiv \sqrt{x^T K(\bar{\mu}) x} \)
Compute one POD basis for each RHS \( i = 1, \ldots, n_{\text{RHS}} \)

\[
\Phi_i(n) \equiv [\phi_1^i, \ldots, \phi_n^i]
\]

Key properties

1. Optimal ordering
   
   - First \( n \) POD basis vectors span an optimal \( n \)-dimensional subspace

2. \( K(\bar{\mu}) \)-orthonormality

\[
\Phi_i(n)^T K(\bar{\mu}) \Phi_i(n) = I
\]

   - \( \Phi_i(n)^T K(\mu) \Phi_i(n) \approx I \) for \( \mu \) near \( \bar{\mu} \)
Three stages to compute approximation $\tilde{u}_i$ at $\mu^{(k)}$ near $\bar{\mu}$

1. Directly solve $n_1$-dimensional reduced equations ($n_1$ small)

$$\Phi_i(n_1)^T K(\mu^{(k)}) \Phi_i(n_1) \hat{u} = \Phi_i(n_1)^T f_i(\mu^{(k)}),$$

$$\tilde{u}_{i,1} = \Phi(n_1) \hat{u}$$

- Accurate (Property 1) and low cost ($n_1$ small)

2. Iteratively solve $n_2$-dimensional reduced equations ($n_2 \gg n_1$)

$$\Phi_i(n_2)^T K(\mu^{(k)}) \Phi_i(n_2) \hat{u} = \Phi_i(n_2)^T \left( f_i(\mu^{(k)}) - K(\mu^{(k)}) \tilde{u}_{i,1} \right),$$

$$\tilde{u}_{i,2} = \tilde{u}_{i,1} + \Phi_i(n_2) \hat{u}$$

- Use augmented CG without forming reduced matrix
- More accurate (Property 1) and low cost (Property 2)
3. Iteratively solve full state equations to specified tolerance $\varepsilon_k$:

$$K(\mu^{(k)})\hat{u} = f_i(\mu^{(k)}) - K(\mu^{(k)})\tilde{u}_{i,2}$$

$$\tilde{u}_i = \tilde{u}_{i,2} + \hat{u}$$

- Use augmented PCG (Farhat et al., 1994)
- Provides “adaptivity” to meet any specified tolerance
- Preconditioner: incomplete Cholesky, previous stiffness (Kirsch, 2002)
- Multiple-RHS (solving state equations + sensitivity analysis)
  - Sequentially execute Stages 1-3 for $i = 1, \ldots, n_{\text{RHS}}$
  - Stage 1 approximation space includes search directions from all previous RHS
Optimization with POD-Krylov ROM

- Optimization procedure

1. Calibrate ROM by high-fidelity simulation
2. Optimize using POD-Krylov ROM of desired accuracy

- Fewer high-fidelity simulations needed
- Accumulate snapshots → POD continually improves (cheaper)
Finite element model with 56,916 degrees of freedom

13 design variables (5 shape, 8 material)

Example: V-22 Osprey wing panel
Problem Statement

- Given: 10 previously-queried designs and 2 new designs

Compute: approximations $\tilde{u}_i, \ i = 1, \ldots, n_{\text{RHS}}$ satisfying

$$\frac{\| f_i(\mu) - K(\mu)\tilde{u}_i \|_2}{\| f_i(\mu) \|_2} < 10^{-2}$$

at the new designs
Error convergence

\( n_{\text{RHS}} = 1 \)

- End of POD approximation

<table>
<thead>
<tr>
<th>Simulation type</th>
<th>( n_{\text{RHS}} )</th>
<th>Speedup (flops), Design A</th>
<th>Speedup (flops), Design B</th>
</tr>
</thead>
<tbody>
<tr>
<td>State equations</td>
<td>1</td>
<td>2.33</td>
<td>7.30</td>
</tr>
<tr>
<td>State equations + direct sensitivity</td>
<td>14</td>
<td>1.78</td>
<td>1.71</td>
</tr>
</tbody>
</table>

Carlberg and Farhat

POD-Based Iterative Solver
Conclusions

- A novel adaptive POD-Krylov reduced order model
  - Compute approximations of any desired accuracy
  - Efficiency due to choice of POD snapshots, weights, and norm
  - 1.7x to 7.3x speedup over existing iterative methods
  - Anticipate at least 3x faster structural design optimizations

- Future work
  - Implement within an optimization algorithm
  - Combine with other augmented Krylov approaches (deflation)
  - Extend to systems with non-SPD matrices and domain decomposition problems (FETI)
Thank You!

Questions?