Multifidelity modeling: Exploiting structure in high-dimensional problems

Karen Willcox
Joint work with Tiangang Cui, Andrew March, Youssef Marzouk, Leo Ng

Workshop on Numerical Methods for High-dimensional Problems

Ecole des Ponts Paristech
April 15, 2014
Collaborators and Acknowledgements

- Andrew March: Multifidelity optimization
- Leo Ng: Multifidelity uncertainty quantification
- Tiangang Cui: Statistical inverse problems
- Professor Youssef Marzouk

- AFOSR Computational Mathematics Program: AFOSR MURI on Uncertainty Quantification (F. Fahroo)
- DOE Applied Mathematics Program: DiaMonD Multifaceted Mathematics Integrated Capability Center (S. Landsberg)
• What is multifidelity modeling?
• Motivation
• Multifidelity modeling approaches:
  – Optimization
  – Inverse problems
  – Uncertainty quantification
Multifidelity modeling

Often have available several physical and/or numerical models that describe a system of interest.

– Models may stem from different resolutions, different assumptions, surrogates, approximate models, etc.
– Each model has its own “fidelity” and computational cost

Today’s focus:

– Multifidelity setup with two models: a “truth” full-order model and a reduced-order model
– Want to use the reduced model to accelerate solution of optimization, uncertainty quantification, or inverse problem solution {opt, UQ, inverse}
Projection-based model reduction

\[
\begin{align*}
\dot{x} &= A(p)x + B(p)u \\
y &= C(p)x \\
x &\approx Vx_r \\
r &= V\dot{x}_r - AVx_r - Bu \\
y_r &= CVx_r
\end{align*}
\]

\[
\begin{align*}
A_r(p) &= W^T A(p) V \\
B_r(p) &= W^T B(p) \\
C_r(p) &= C(p) V \\
\dot{x}_r &= A_r(p)x_r + B_r(p)u \\
y_r &= C_r(p)x_r
\end{align*}
\]

\[W^T r = 0\]

\(x \in \mathbb{R}^N: \text{ state vector}\)

\(p \in \mathbb{R}^{N_p}: \text{ parameter vector}\)

\(u \in \mathbb{R}^{N_i}: \text{ input vector}\)

\(y \in \mathbb{R}^{N_o}: \text{ output vector}\)

\(x_r \in \mathbb{R}^n: \text{ reduced state vector}\)

\(V \in \mathbb{R}^{N \times n}: \text{ reduced basis}\)
Why use a multifidelity formulation?

Reduced model (approximate)  Full model ("truth")
Why use a multifidelity formulation?

Reduced model (approximate)

Computationally cheap(er)

Full model ("truth")

Computationally expensive
Why use a multifidelity formulation?

- Replace full model with reduced model and solve \{opt, UQ, inverse\}
- Propagate error estimates on forward predictions to determine error in \{opt, UQ, inverse\} solutions (may be non-trivial)
Why use a multifidelity formulation?

- Replace full model with reduced model and solve \{opt, UQ, inverse\}
- Hope for the best

Reduced model (approximate)

Certified? no

Full model (“truth”)
Why use a multifidelity formulation?

Reduced model (approximate) | Full model ("truth")

Certified? no

- Use a multifidelity formulation that invokes both the reduced model and the full model
- Trade computational cost for the ability to place guarantees on the solution of \{opt, UQ, inverse\}
Why use a multifidelity formulation?

Reduced model (approximate)  

Certified?
no

Full model ("truth")

- Use a multifidelity formulation that invokes both the reduced model and the full model
- Trade computational cost for the ability to place guarantees on the solution of \{\text{opt, UQ, inverse}\}
- **Certify the solution of \{\text{opt, UQ, inverse}\} even in the absence of guarantees on the reduced model itself**
Multifidelity Strategies

• For optimization:
  – adaptive model calibration (corrections)
  – combined with trust region model management

• For statistical inverse problems:
  – adaptive delayed acceptance Markov chain Monte Carlo (MCMC) methods

• For forward propagation of uncertainty:
  – control variates
\[
\min_{x} f(x) \\
\text{s.t.} \quad g(x) \leq 0 \\
\quad h(x) = 0
\]
Design optimization formulation

\[
\begin{align*}
\min_x f(x) \\
\text{s.t. } g(x) &\leq 0 \\
h(x) &= 0
\end{align*}
\]

Design variables \( x \)

Objective \( f(x) \)

Constraints \( g(x), h(x) \)

• Interested in optimization of systems governed by PDEs (constraints and objective evaluation is expensive)
Multifidelity optimization formulation

\[
\begin{align*}
\min_{x} & \quad f(x) \\ 
\text{s.t.} & \quad g(x) \leq 0 \\ 
& \quad h(x) = 0
\end{align*}
\]

Design variables \( x \)
Objective \( f(x) \)
Constraints \( g(x), h(x) \)

Diagram:
- 

- \( f_{hi}, g_{hi}, h_{hi} \)
- \( x \) (design variables)
- \( f_{lo} + \alpha, g_{lo} + \beta, h_{lo} + \gamma \)
- \( x_j \) (correction)
- \( f_{hi}, g_{hi}, h_{hi} \) (hi-fi model correction)
- \( \text{hi-fi model} \)
Multifidelity optimization: Surrogate definition

• Denote a surrogate model of \( f_{\text{high}}(x) \) as \( m(x) \)

• The surrogate model could be:

  1. The **low-fidelity function** (reduced model)
     \[
     m(x) = f_{\text{low}}(x) \approx f_{\text{high}}(x)
     \]

  2. The **sum** of the low-fidelity function and an additive correction
     \[
     m(x) = f_{\text{low}}(x) + e(x) \approx f_{\text{high}}(x)
     \]
     where \( e(x) \) is calibrated to the difference \( f_{\text{high}}(x) - f_{\text{low}}(x) \)

  3. The **product** of a low-fidelity function and a multiplicative correction
     \[
     m(x) = \beta_c(x) f_{\text{low}}(x) \approx f_{\text{high}}(x)
     \]
     where \( \beta_c(x) \) is calibrated to the quotient \( f_{\text{high}}(x) / f_{\text{low}}(x) \)

• Update the correction terms as the optimization algorithm proceeds and additional evaluations of \( f_{\text{high}}(x) \) become available
Multifidelity optimization: Trust-region model management

- At iteration $k$, define a trust region centered on iterate $x_k$ with size $\Delta_k$
  \[ B_k = \{ x : \| x - x_k \| \leq \Delta_k \} \]

- $m_k$ is the surrogate model on the $k$th iteration

- Determine a trial step $s_k$ at iteration $k$, by solving a subproblem of the form:

\[
\min_{s_k} \quad m_k(x_k + s_k) \\
\text{s.t.} \quad \|s_k\| \leq \Delta_k
\]

(unconstrained case)
Multifidelity optimization: Trust-region model management

- Evaluate the function at the trial point: $f_{\text{high}}(x_k+s_k)$

- Compute the ratio of the actual improvement in the function value to the improvement predicted by the surrogate model:

$$\rho_k = \frac{f_{\text{high}}(x_k) - f_{\text{high}}(x_k + s_k)}{m_k(x_k) - m_k(x_k + s_k)}$$

- Accept or reject the trial point and update trust region size according to (typical parameters):

<table>
<thead>
<tr>
<th>$\rho^k$</th>
<th>Accept step</th>
<th>Reject step</th>
<th>$\Delta^{k+1}$</th>
</tr>
</thead>
<tbody>
<tr>
<td>$\leq 0$</td>
<td>Reject step</td>
<td>$\Delta^{k+1} \equiv 0.5\Delta^k$</td>
<td></td>
</tr>
<tr>
<td>$0 &lt; \rho^k \leq 0.1$</td>
<td>Accept step</td>
<td>$\Delta^{k+1} \equiv 0.5\Delta^k$</td>
<td></td>
</tr>
<tr>
<td>$0.1 &lt; \rho^k &lt; 0.75$</td>
<td>Accept step</td>
<td>$\Delta^{k+1} \equiv \Delta^k$</td>
<td></td>
</tr>
<tr>
<td>$0.75 \leq \rho^k$</td>
<td>Accept step</td>
<td>$\Delta^{k+1} \equiv 2\Delta^k$</td>
<td></td>
</tr>
</tbody>
</table>
Trust-Region Algorithm for Iteration $k$

1. Compute a step, $s_k$, by solving the trust-region subproblem,

$$\min_{s_k} m_k(x_k + s_k)$$

s.t. $\|s_k\| \leq \Delta_k$.

2. Evaluate $f_{\text{high}}(x_k + s_k)$.

3. Compute the ratio of actual improvement to predicted improvement,

$$\rho_k = \frac{f_{\text{high}}(x_k) - f_{\text{high}}(x_k + s_k)}{m_k(x_k) - m_k(x_k + s_k)}.$$

4. Accept or reject the trial point according to $\rho_k$,

$$x_{k+1} = \begin{cases} x_k + s_k & \text{if } \rho_k > 0 \\ x_k & \text{otherwise}. \end{cases}$$

5. Update the trust region size according to $\rho_k$,

$$\Delta_{k+1} = \begin{cases} \gamma_1 \Delta_k & \text{if } \rho_k \leq \eta_1 \\ \Delta_k & \text{if } \eta_1 < \rho_k < \eta_2 \\ \gamma_2 \Delta_k & \text{if } \rho_k \geq \eta_2. \end{cases}$$
Trust-Region Demonstration
Trust-region model management: Corrections and convergence

• Provably convergent to local minimum of high-fidelity function if surrogate is first-order accurate at center of trust region [Alexandrov et al., 2001]

• Additive correction:  \[ a(x) = f_{\text{high}}(x) - f_{\text{low}}(x) \]

with surrogate constructed as

\[ m_k(x) = f_{\text{low}}(x) + a(x_k) + \nabla a(x_k)^T (x - x_k) \]

• Multiplicative correction:  \[ \beta(x) = \frac{f_{\text{high}}(x)}{f_{\text{low}}(x)} \]

with surrogate constructed as

\[ m_k(x) = [\beta(x_k) + \nabla \beta(x_k)^T (x - x_k)] f_{\text{low}}(x) \]

• Only first-order corrections required to guarantee convergence; quasi-second-order corrections accelerate convergence [Eldred et al., 2004]

• Trust-region POD [Arian, Fahl, Sachs, 2000]
• Derivative-free trust region approaches
  [Conn, Scheinberg, and Vicente, 2009]

• Provably convergent under appropriate conditions if the surrogate model is “fully linear”

\[
\left\| \nabla f_{\text{high}}(\mathbf{x}) - \nabla m_k(\mathbf{x}) \right\| \leq \kappa_g \Delta_k
\]

\[
\left| f_{\text{high}}(\mathbf{x}) - m_k(\mathbf{x}) \right| \leq \kappa_f \Delta_k^2
\]

• Achieved through adaptive corrections or adaptive calibration e.g., radial basis function calibration with sample points chosen to make surrogate model fully linear by construction
  [Wild, Regis and Shoemaker, 2011; Wild and Shoemaker, 2013]

• Key: never need gradients wrt the high-fidelity model
Multifidelity design optimization example: Aircraft wing (with black-box codes)

Design variables: wing geometry, structural members

Objectives: weight, lift-to-drag ratio

Disciplines: aerodynamics, structures

Aerodynamics and structures exchange pressure loading and deflections, requiring an iterative solve for each analysis.

Multifidelity models:

Structures: Nastran (commercial finite element code; MSC)  
Beam model

Aerodynamics: Panair (panel code for inviscid flows; NASA)  
FRICTION (skin friction and form factors; W. Mason)  
AVL (vortex-lattice model; M. Drela)  
Kriging surrogate
Multifidelity design optimization example: Aircraft wing

Multifidelity approach:

• Trust region model management
  – Derivative free framework [Conn et al., 2009]

• Adaptive calibration of surrogates
  – Radial basis function calibration to provide fully linear models [Wild et al., 2009]
  – Calibration applied to correction function (difference between high- and low-fidelity models) [Kennedy & O’Hagan, 2001]

• Computational speed-up + robustness to code failures

<table>
<thead>
<tr>
<th>Low-Fidelity Model</th>
<th>Nastran Evals.</th>
<th>Panair Evals.</th>
<th>Time* (days)</th>
</tr>
</thead>
<tbody>
<tr>
<td>None</td>
<td>7,425</td>
<td>7,425</td>
<td>4.73</td>
</tr>
<tr>
<td>AVL/Beam Model</td>
<td>5,412</td>
<td>5,412</td>
<td>3.45</td>
</tr>
<tr>
<td>Kriging Surrogate</td>
<td>3,232</td>
<td>3,232</td>
<td>2.06</td>
</tr>
</tbody>
</table>

*Time corresponds to average of 30s per Panair evaluation, 25s per Nastran evaluation, and serial analysis of designs within a discipline.
\[ \pi(x|d) \sim L(d|x) \pi_0(x) \]
Large-scale statistical inverse problems

- Data are limited in number, noisy, and indirect
- State-space is high dimensional (PDE model)
- Unknown parameters are high-dimensional
Large-scale statistical inverse problems

Bayes rule:

\[ \pi(x|d) \sim L(d|x) \pi_0(x) \]

- posterior
- likelihood
- prior
Large-scale statistical inverse problems: Exploiting low-rank structure

Bayes rule: \[ \pi(x|d) \sim L(d|x) \pi_0(x) \]
- posterior
- likelihood
- prior

• Low-rank structure in the state space: Data-driven model reduction [Cui, Marzouk, W., 2014]

• Low-rank structure in the parameter space: Efficient posterior exploration (likelihood-induced subspace) [Lieberman, W., 2010; Cui, Martin, Marzouk, 2014]
Exploring the posterior: MCMC Sampling

Markov chain Monte Carlo (MCMC) methods: black box but expensive ways to sample the posterior $\pi(x|d)$ [Metropolis et al., 1953; Hastings, 1970]

- Requires many (many) iterations to generate enough samples to characterize the posterior
- Many samples are rejected

\[
\alpha(x_n, y) = \min \left[ 1, \frac{\pi(y|d)q(y, x_n)}{\pi(x_n|d)q(x_n, y)} \right]
\]

If Uniform[0, 1] < $\alpha(x_n, y)$

Expensive forward model solve
Multifidelity: Adaptive delayed acceptance MCMC sampling

- Sampling of the exact posterior is guaranteed by the second stage [Chen & Liu, 1998; Christen & Fox, 2005]
- Speed-up: not all samples are evaluated by full model
Adaptive reduced models for multifidelity inference

• Reduced model is evaluated from “snapshots” (solutions at selected parameter values)

• These evaluations are used to construct the reduced basis

• Standard approach: snapshots are selected offline from the prior (e.g., Wang and Zabaras, 2004; Lieberman et al., 2010)

• We propose a data-driven adaptive approach using delayed acceptance:
  to provide a formal framework to manage use of the ROM (multifidelity)

  and to adaptively select snapshots and update the ROM on the fly
Simultaneous model reduction and posterior exploration

• Suppose we have a reduced model constructed from an initial reduced basis

• Stage 1:
  – At each MCMC iteration, first sample the approximate posterior distribution ($\pi^*$) based on the reduced model for $m$ steps using a standard Metropolis-Hasting algorithm
  – Decreases the sample correlation with low computational cost by simulating an approximate Markov chain [Cui, 2010]

• Stage 2:
  – The last state of the Stage 1 Markov chain is the proposal candidate
  – Compute acceptance probability ($\alpha$) based on full posterior density value (ensures that we sample the exact posterior)
  – After each full posterior density evaluation, the state of the associated forward model evaluation is a potential new snapshot
Simultaneous model reduction and posterior exploration

From $x_n$, sampling $\pi^*$ for $m$ iterations

Compute the error $t(x_{n+m})$

- Compute the error of the reduced model output estimate at each new posterior sample
- Update the reduced basis with the new snapshot when the error exceeds a threshold $\epsilon$
- The resulting reduced model is data-driven, since it uses the information provided by the observed data (in the form of the posterior distribution) to select samples for computing the snapshots
Simultaneous model reduction and posterior exploration

From $x_n$, sampling $\pi^*$ for $m$ iterations

Compute the error $t(x_{n+m})$

- Can also use error estimator ($\hat{t}$) (e.g., dual weighted residual [Meyer, Matthies 2003]) but then we lose the strong guarantee of sampling the exact posterior

Full Target

Approximate

If $|t| > \epsilon$, update ROM
Iterate forward

If $|\hat{t}| > \epsilon$, evaluate $\pi$, and update ROM
Otherwise, $\pi^* \approx \pi$
Iterate forward
Inverse problem example: 9D test case

\[-\nabla \cdot (k(r) \nabla u(r)) = f(r), \quad r \in D\]

\[k(r) \nabla u(r) \cdot \vec{n}(r) = 0, \quad r \in \partial D\]

In the domain $r \in [0, 1]^2$, try to infer the diffusivity

\[k(r) = \sum_{i=1}^{9} b_i(r)x_i\]

\[\log(x_i) \sim \mathcal{N}(\mu_i, \sigma_i^2)\]

121 potential measurements, signal to noise ratio 50.

Full model has $120 \times 120$ elements.
### Inverse problem example: Sampling efficiency

<table>
<thead>
<tr>
<th></th>
<th>Reference</th>
<th>Full Target</th>
<th>Approximate</th>
</tr>
</thead>
<tbody>
<tr>
<td>Error threshold $\epsilon$</td>
<td>-</td>
<td>$10^{-1}$</td>
<td>$10^{-2}$</td>
</tr>
<tr>
<td>Basis vectors</td>
<td>-</td>
<td>14</td>
<td>33</td>
</tr>
<tr>
<td>ESS / CPU time</td>
<td>0.058</td>
<td>2.5</td>
<td>2.7</td>
</tr>
<tr>
<td>Speed-up factor</td>
<td>1</td>
<td>43</td>
<td>46</td>
</tr>
</tbody>
</table>

- Run both algorithms for $5 \times 10^5$ iterations, with $\epsilon = 10^{-1}, 10^{-2}, 10^{-3}$.
- $\epsilon$ is normalized by the standard derivation measurement noise.
- A reference MCMC (only based on the full model) is simulated for $5 \times 10^5$ iterations.
- Speed-up factor is estimated from CPU time per effective sample.
Inverse problem example: Sampling accuracy

- **Reference**: 9.6 hours
- **Exact ($\epsilon = 10^{-1}$)**: 15 minutes
- **Approx ($\epsilon = 10^{-1}$)**: 2.3 minutes
For benchmarking, $10^4$ snapshots from the prior to construct the ROM.

- The data-driven ROM are built with $\epsilon = 10^{-3}$.
- The true error for both ROMs are calculated on $10^4$ posterior samples.
- The true error is normalized by the standard derivation of measurement noise.
Inverse problem example: A high-dimensional case

The diffusivity $k(r) = \exp[\chi(r)]$ is spatially distributed, has the prior

$$\chi(r) \sim \mathcal{N}(\mu, \mathcal{C}), \quad \text{corr}(r, s) = \exp \left[-\left( \frac{|r - s|}{L} \right)^2 \right]$$

We use $L = 0.25$, and apply Karhunen-Loéve expansion to $\mathcal{C}$. We keep the first 45 modes $\sim 99.99\%$ energy.
**Inverse problem example: Sampling efficiency**

<table>
<thead>
<tr>
<th></th>
<th>Reference</th>
<th>Full Target</th>
<th>Approximate</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Error threshold $\epsilon$</strong></td>
<td>-</td>
<td>$10^{-1}$</td>
<td>$10^{-1}$</td>
</tr>
<tr>
<td><strong>Basis vectors</strong></td>
<td>-</td>
<td>64</td>
<td>62</td>
</tr>
<tr>
<td><strong>ESS / CPU time</strong></td>
<td>0.033</td>
<td>2.2</td>
<td>8.2</td>
</tr>
<tr>
<td><strong>Speed-up factor</strong></td>
<td>1</td>
<td>67</td>
<td>249</td>
</tr>
</tbody>
</table>

- Run both algorithms for $5 \times 10^5$ iterations.
- A reference MCMC (only based on the full model) is simulated for $5 \times 10^5$ iterations.
- Speed-up factor is estimated from CPU time per effective sample.
Inverse problem example: Sampling accuracy

Reference

Exact, $\epsilon = 10^{-1}$

Approx., $\epsilon = 10^{-1}$

Mean

Standard deviation

21 hours

17 minutes

5 minutes
\[
\min_x f(x, s(x)) \\
\text{s.t. } g(x, s(x)) \leq 0 \\
h(x, s(x)) = 0
\]
The challenge of optimization under uncertainty (OUU)

\[
\begin{align*}
\min_{x} & \quad f(x, s(x)) \\
\text{s.t.} & \quad g(x, s(x)) \leq 0 \\
& \quad h(x, s(x)) = 0
\end{align*}
\]

- Design variables $x$
- Uncertain parameters $u$
- Model outputs $y(x, u)$
- Statistics of model $s(x)$

High-fidelity model embedded in a UQ loop in an optimization loop
- Large computational cost
- Need an optimizer that is tolerant to noisy estimates of statistics
Multifidelity optimization under uncertainty

\[ \min_x f(x, s(x)) \]
\[ \text{s.t. } g(x, s(x)) \leq 0 \]
\[ h(x, s(x)) = 0 \]

Design variables \( x \)
Uncertain parameters \( u \)
Model outputs \( y(x, u) \)
Statistics of model \( s(x) \)

Diagram:
- **optimizer**
- **UQ**
- **hi-fi model**
Multifidelity OUU approach: Control variates

Minimize \( f(x, s(x)) \) subject to:
\[
\begin{align*}
  g(x, s(x)) & \leq 0 \\
  h(x, s(x)) & = 0
\end{align*}
\]

**Design variables** \( x \)
**Uncertain parameters** \( u \)
**Model outputs** \( y(x, u) \)
**Statistics of model** \( s(x) \)

Control variates: Exploit model correlation
- Estimate correlation between high- and low-fidelity models
- Related to multilevel Monte Carlo (Giles, 2008; Speight, 2009)
- RB models also used with control variates in Boyaval & Lelièvre, 2010
Problem setup

- **Design variables**
- **Random uncertain parameters**

The diagram illustrates the relationship between the design variables $x$, the random uncertain parameters $U$, the high-fidelity model $f_{\text{high}}(x, U)$, and the low-fidelity model $f_{\text{low}}(x, U)$. The outputs $A$ and $B$ are the random outputs of the high- and low-fidelity models, respectively.

For the random uncertain parameters $U$:

- $u_i = \text{samples of } U$
- $a_i = f_{\text{high}}(x, u_i) = \text{samples of } A$
- $b_i = f_{\text{low}}(x, u_i) = \text{samples of } B = a_i + \text{error}$

For the statistics of $A$:

- $s_A = \text{statistics of } A$ (e.g., mean, variance)
- $\hat{s}_A = \text{estimator of } s_A$

The problem setup includes:

\[
\min_x f(x, s_A(x)) \quad \text{approximated by} \quad \min_x f(x, \hat{s}_A)
\]

subject to:

\[
\begin{align*}
g(x, s_A(x)) &\leq 0 \\
g(x, \hat{s}_A(x)) &\leq 0
\end{align*}
\]
Variance reduction with control variate

- Regular MC estimator for $s_A = \mathbb{E}[A]$ using $n$ samples of $A$:

  $$\bar{a}_n = \frac{1}{n} \sum_{i=1}^{n} a_i \quad \text{Var}[\bar{a}_n] = \frac{\sigma_A^2}{n}$$

- Control variate (CV) estimator of $s_A$:
  - Additional random variable $B$ with known $s_B = \mathbb{E}[B]$:
    $$\hat{s}_A = \bar{a}_n + \alpha (s_B - \bar{b}_n)$$
    $$\text{Var}[\hat{s}_A] = \frac{\sigma_A^2 + \alpha^2 \sigma_B^2 - 2\alpha \rho_{AB} \sigma_A \sigma_B}{n}$$

- Minimize $\text{Var}[\hat{s}_A]$ with respect to $\alpha$:
  $$\text{Var}[\hat{s}_A^*] = (1 - \rho_{AB}^2) \frac{\sigma_A^2}{n} \leq 1$$

Definitions:

- $\sigma_A^2 = \text{Var}[A]$
- $\sigma_B^2 = \text{Var}[B]$
- $\rho_{AB} = \text{Corr}[A, B]$
Low-fidelity model as control variate

- Multifidelity estimator of \( s_A \) based on control variate method:
  - \( A = \) random output of high-fidelity model
  - \( B = \) random output of low-fidelity model (\( s_B \) unknown)

\[
\hat{s}_{A,p} = \bar{a}_n + \alpha (\bar{b}_m - \bar{b}_n) \quad \text{with} \quad m \gg n
\]

\[
\text{Var}[\hat{s}_{A,p}] = \frac{\sigma_A^2 + \alpha^2 \sigma_B^2 - 2\alpha \rho_{AB} \sigma_A \sigma_B}{n} - \frac{\alpha^2 \sigma_B^2 - 2\alpha \rho_{AB} \sigma_A \sigma_B}{m}
\]

- Using difference \( (\bar{b}_m - \bar{b}_n) \) as correction to \( \bar{a}_n \)

- Leveraging correlation between \( A \) and \( B \)
  - Correlation captured in \( \alpha \)

Definitions:

\[
\sigma_A^2 = \text{Var}[A] \\
\sigma_B^2 = \text{Var}[B] \\
\rho_{AB} = \text{Corr}[A, B]
\]
Computational budget allocation

- Define computational effort $p$ as **equivalent** # of high-fidelity model evaluations

$$p = n + \frac{m}{w} = n \left(1 + \frac{r}{w}\right) \quad \text{where} \quad r = \frac{m}{n} \quad \text{and} \quad w = \frac{\text{high-fidelity evaluation time}}{\text{low-fidelity evaluation time}}$$

- For fixed $p$, minimize $\text{Var}[\hat{S}_{A,p}]$ with respect to $\alpha$ and $r$

$$\alpha^* = \rho_{AB} \frac{\sigma_A}{\sigma_B} \quad \quad r^* = \frac{w \rho_{AB}^2}{\sqrt{1 - \rho_{AB}^2}} \quad \quad \text{Var}[\hat{S}_{A,p}] = \left[1 - \left(1 - \frac{1}{r^*}\right) \rho_{AB}^2\right] \left(1 + \frac{r^*}{w}\right) \frac{\sigma_A^2}{p}$$

- Limiting cases:

  (i) Low-fidelity model “free”: as $w \to \infty$, then $\text{Var}[\hat{S}_{A,p}] \to (1 - \rho_{AB}^2) \frac{\sigma_A^2}{p}$

  (ii) Low-fidelity model “perfect”: as $\rho_{AB} \to 1$, then $\text{Var}[\hat{S}_{A,p}] \to \frac{1}{w} \frac{\sigma_A^2}{p}$

**Definitions:** $\sigma_A^2 = \text{Var}[A], \sigma_B^2 = \text{Var}[B], \rho_{AB} = \text{Corr}[A, B]$
Model correlation over design space

- What if low-fidelity model unavailable?
  - Use $M_{\text{high}}(x + \Delta x, U)$ as **surrogate** for $M_{\text{high}}(x, U)$

- At current design point $x_k$
  - Define $A = M_{\text{high}}(x_k, U)$
  - Want to compute $\hat{s}_A$ as estimator of $s_A = \mathbb{E}[A]$  

- Previously visited design point $x_\ell$ where $\ell < k$
  - Define surrogate as $C = M_{\text{high}}(x_\ell, U)$
  - Reuse available data: $\hat{s}_C$ as estimator of $s_C = \mathbb{E}[C]$ with error $\text{Var}[\hat{s}_C]$

---

**Diagram**

<table>
<thead>
<tr>
<th>design variables</th>
<th>estimators</th>
</tr>
</thead>
<tbody>
<tr>
<td>$x_\ell$</td>
<td>$\hat{s}<em>A(x</em>\ell)$</td>
</tr>
<tr>
<td>$x_{k-1}$</td>
<td>$\hat{s}<em>A(x</em>{k-1})$</td>
</tr>
<tr>
<td>$x_k$</td>
<td>$\hat{s}_A(x_k)$</td>
</tr>
</tbody>
</table>

---

**Information Reuse Estimator**
Acoustic horn example

- Helmholtz equation for propagation of acoustic waves through 2-D horn
  - High-fidelity model: Finite element model (FEM) with 35,895 states
  - Low-fidelity model I: Reduced basis model (RBM) with \( N = 25 \) states
  - Low-fidelity model II: Reduced basis model (RBM) with \( N = 30 \) states
  - Ratio of evaluation cost \( w = 40 \)

Input: wave number \( K \sim \) uniform

Input: upper horn wall impedance \( Z_u \sim \) normal

Input: lower horn wall impedance \( Z_l \sim \) normal

Output: reflection coefficient, \( S_r \)

Acoustic horn models due to D.B.P. Huynh
Acoustic horn example – uncertainty propagation

- $w = 40$ in both cases
- Correlation between FEM and
  - RBM (N = 25) $\approx 0.928$
  - RBM (N = 30) $\approx 0.996$
- Increasing correlation increases efficiency of multifidelity estimator
Acoustic horn example – uncertainty propagation

- Apply regular MC simulation directly to reduced basis model?
  - Bias of the low-fidelity model cannot be reduced regardless of # of samples used
  - Multifidelity MC simulation can achieve arbitrarily small error tolerance

- “Good” low-fidelity model based on correlation, not difference in outputs
Acoustic horn example – robust optimization

**Decision variables:** horn geometry, $b$

**Uncertainty:** wavenumber, wall impedances

**Output of interest:** reflection coefficient, $s_r$

$$\min_b \mathbb{E}[s_r] + \sqrt{\text{Var}[s_r]}$$

---

<table>
<thead>
<tr>
<th>Method</th>
<th>Equivalent number of hi-fi evaluations</th>
</tr>
</thead>
<tbody>
<tr>
<td>Regular MC</td>
<td>44,343</td>
</tr>
<tr>
<td>Multifidelity MC</td>
<td>6,979 (-84%)</td>
</tr>
</tbody>
</table>

**Optimization algorithm:** Implicit filtering [Kelley, 2011]
Example: High-fidelity wing optimization

- Shape optimization of (roughly) Bombardier Q400 wing
  - Free-form deformation geometry control [Kenway et al. 2010]

- Coupled aerostructural solver [Kennedy and Martins 2010]
  - Aerodynamics: TriPan panel method
  - Structures: Toolkit for the Analysis of Composite Structures (TACS) finite element method

<table>
<thead>
<tr>
<th></th>
<th>Coarse</th>
<th>Fine</th>
</tr>
</thead>
<tbody>
<tr>
<td>Aerodynamic Panels</td>
<td>1000</td>
<td>2960</td>
</tr>
<tr>
<td>Structural d.o.f.</td>
<td>5624</td>
<td>14,288</td>
</tr>
<tr>
<td>Eval time</td>
<td>6 s</td>
<td>24 s</td>
</tr>
</tbody>
</table>
High-fidelity wing optimization

- 46 design variables:
  - 8 wing twist angles, 19 forward spar thicknesses, 19 aft spar thicknesses

- 7 random inputs:
  - Take-off weight, Mach number, material properties (density, elastic modulus, Poisson ratio, yield stress), wing weight fraction

- Objective = drag (formulated as mean + 2 std)

- 4 nonlinear stress constraints (formulated as mean + 2 std ≤ 0)

- 36 linear geometry constraints (deterministic)

- Optimization loop: COBYLA constrained derivative-free solver [Powell 1994]

- Simulation loop: Fixed RMSE for estimators specified, number of samples allowed to vary
High-fidelity wing optimization

- Solved on 16-processor desktop machine
- Combined estimator enable OUU solution in reasonable turnaround time
- Regular Monte Carlo estimator would take about 3.2 months

<table>
<thead>
<tr>
<th></th>
<th>Computational Effort</th>
<th>Total Time (days)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Regular MC</td>
<td>--</td>
<td>--</td>
</tr>
<tr>
<td>Info Reuse</td>
<td>$7 \times 10^4$</td>
<td>13.4</td>
</tr>
<tr>
<td>Combined</td>
<td>$5 \times 10^4$</td>
<td>9.7</td>
</tr>
</tbody>
</table>
Certified?

- Use a multifidelity formulation that invokes both the reduced model and the full model
- Trade computational cost for the ability to place guarantees on the solution of \{opt, UQ, inverse\}
- **Certify the solution of \{opt, UQ, inverse\}** even in the absence of guarantees on the reduced model itself
Conclusions

“All models are wrong, but some are useful.”

George Box, 1979

• A formal framework for multifidelity modeling can
  – help us understand when our (reduced) models are useful
  – provide a responsible way to use our wrong-but-useful models for optimization, inversion, and uncertainty quantification

• Towards a richer definition of fidelity:
  – In almost all existing multifidelity methods, “fidelity” = a linear ranking of models, with some “high-fidelity” model denoted as “truth”
  – In practice, the relationship between models and reality—and among different sources of information—is much richer than just a ranking
  – Models and/or experiments they tell us different things about the design problem, with the collective information they provide being greater than the individual parts