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)

Outline

- 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

$$\begin{aligned}\dot{\mathbf{x}} &= \mathbf{A}(\mathbf{p})\mathbf{x} + \mathbf{B}(\mathbf{p})\mathbf{u}\\ \mathbf{y} &= \mathbf{C}(\mathbf{p})\mathbf{x}\end{aligned}$$

$\mathbf{x} \approx \mathbf{V}\mathbf{x}_r$ →

$$\begin{aligned}\mathbf{r} &= \mathbf{V}\dot{\mathbf{x}}_r - \mathbf{A}\mathbf{V}\mathbf{x}_r - \mathbf{B}\mathbf{u}\\ \mathbf{y}_r &= \mathbf{C}\mathbf{V}\mathbf{x}_r\end{aligned}$$

↓ $\mathbf{W}^T\mathbf{r} = 0$

$$\begin{aligned}\dot{\mathbf{x}}_r &= \mathbf{A}_r(\mathbf{p})\mathbf{x}_r + \mathbf{B}_r(\mathbf{p})\mathbf{u}\\ \mathbf{y}_r &= \mathbf{C}_r(\mathbf{p})\mathbf{x}_r\end{aligned}$$

$$\begin{aligned}\mathbf{A}_r(\mathbf{p}) &= \mathbf{W}^T\mathbf{A}(\mathbf{p})\mathbf{V}\\ \mathbf{B}_r(\mathbf{p}) &= \mathbf{W}^T\mathbf{B}(\mathbf{p})\\ \mathbf{C}_r(\mathbf{p}) &= \mathbf{C}(\mathbf{p})\mathbf{V}\end{aligned}$$

- $\mathbf{x} \in \mathbf{R}^N$: state vector
- $\mathbf{p} \in \mathbf{R}^{N_p}$: parameter vector
- $\mathbf{u} \in \mathbf{R}^{N_i}$: input vector
- $\mathbf{y} \in \mathbf{R}^{N_o}$: output vector
- $\mathbf{x}_r \in \mathbf{R}^n$: reduced state vector
- $\mathbf{V} \in \mathbf{R}^{N\times n}$: reduced basis

Why use a multifidelity formulation?

Why use a multifidelity formulation?

Reduced model (approximate)

Full model ("truth")

Computationally cheap(er)

Computationally expensive

Why use a multifidelity formulation?

Reduced model
(approximate)

Full model
("truth")

Certified?

yes

- 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?

Reduced model
(approximate)

Full model
("truth")

Certified?

no

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

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)

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}
- Certify the solution of {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

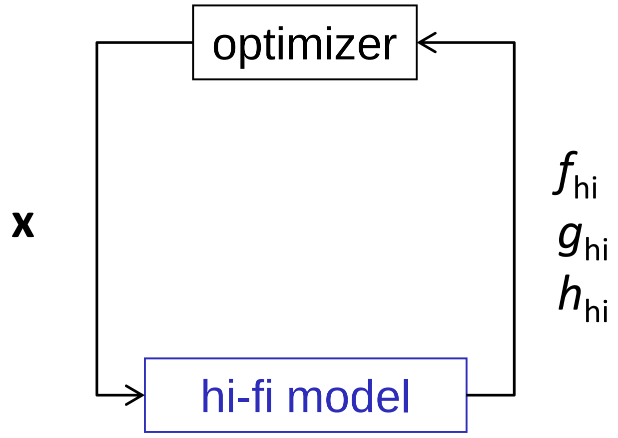
$$\min_{x} f(x)$$
$$\text{s.t.} \quad g(x) \leq 0$$
$$h(x) = 0$$

OPTIMIZATION

Design optimization formulation

$$\min_{x} f(x)$$
$$\text{s.t.} \quad g(x) \leq 0$$
$$h(x) = 0$$

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

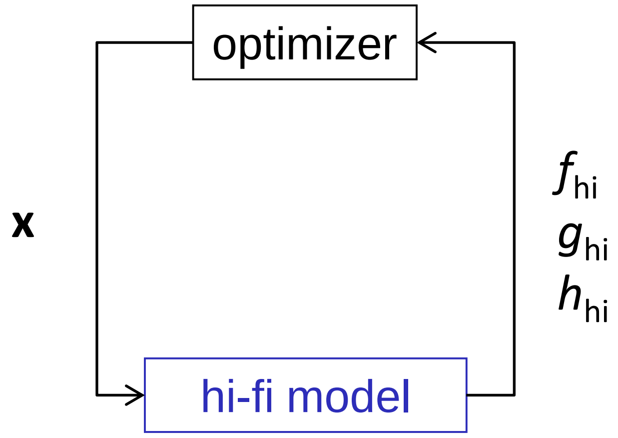


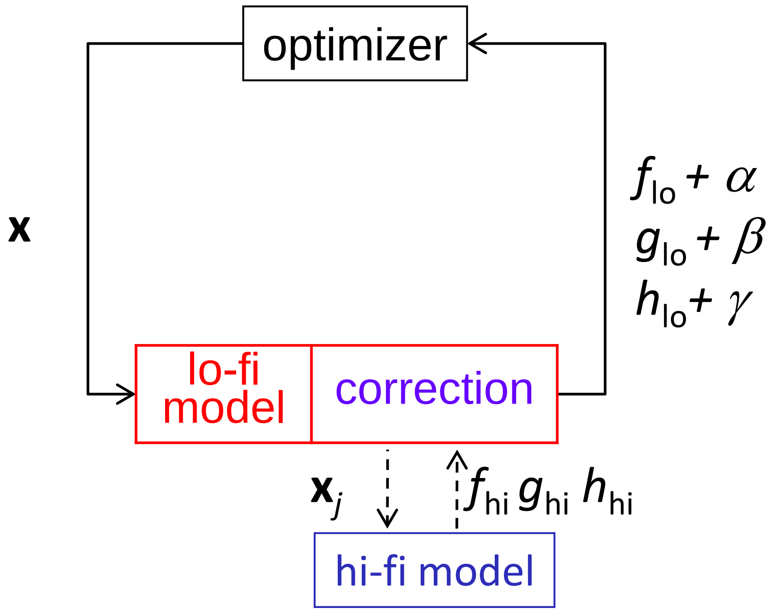
- Interested in optimization of systems governed by PDEs (constraints and objective evaluation is expensive)

Multifidelity optimization formulation

$$\min_{x} f(x)$$
$$\text{s.t.} \quad g(x) \leq 0$$
$$h(x) = 0$$

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





Multifidelity optimization: Surrogate definition

- Denote a surrogate model of $f_{\text{high}}(\mathbf{x})$ as $m(\mathbf{x})$
- The surrogate model could be:
  1. The **low-fidelity function** (reduced model)
  $$m(\mathbf{x}) = f_{\text{low}}(\mathbf{x}) \approx f_{\text{high}}(\mathbf{x})$$
  2. The **sum** of the low-fidelity function and an additive correction
  $$m(\mathbf{x}) = f_{\text{low}}(\mathbf{x}) + e(\mathbf{x}) \approx f_{\text{high}}(\mathbf{x})$$
  where $e(\mathbf{x})$ is calibrated to the difference $f_{\text{high}}(\mathbf{x})$- $f_{\text{low}}(\mathbf{x})$
  3. The **product** of a low-fidelity function and a multiplicative correction
  $$m(\mathbf{x}) = \beta_c(\mathbf{x}) f_{\text{low}}(\mathbf{x}) \approx f_{\text{high}}(\mathbf{x})$$
  where $\beta_c(\mathbf{x})$ is calibrated to the quotient $f_{\text{high}}(\mathbf{x}) / f_{\text{low}}(\mathbf{x})$
- Update the correction terms as the optimization algorithm proceeds and additional evaluations of $f_{\text{high}}(\mathbf{x})$ become available

Multifidelity optimization: Trust-region model management

- At iteration k, define a trust region centered on iterate $\mathbf{x}_k$ with size Δ_k

$$\mathcal{B}_k = \{\mathbf{x} : \|\mathbf{x} - \mathbf{x}_k\| \leq \Delta_k\}$$

- m_k is the surrogate model on the kth iteration

- Determine a trial step $\boldsymbol{s}_k$ at iteration k, by solving a subproblem of the form:

$$\min_{\mathbf{x}} \; f(\mathbf{x}) \quad \Longrightarrow \quad \boxed{\begin{aligned} \min_{\mathbf{s}_k} \quad & m_k(\mathbf{x}_k + \mathbf{s}_k) \\ \text{s.t.} \quad & \|\mathbf{s}_k\| \leq \Delta_k \end{aligned}}$$

(unconstrained case)

Multifidelity optimization: Trust-region model management

- Evaluate the function at the trial point: $f_{\text{high}}(\mathbf{x}_k+\mathbf{s}_k)$

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

$$\rho_k = \frac{f_{\text{high}}(\mathbf{x}_k) - f_{\text{high}}(\mathbf{x}_k + \mathbf{s}_k)}{m_k(\mathbf{x}_k) - m_k(\mathbf{x}_k + \mathbf{s}_k)}$$

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

$\rho^k \leq 0$	Reject step	$\Delta^{k+1} \equiv 0.5\Delta^k$
$0 < \rho^k \leq 0.1$	Accept step	$\Delta^{k+1} \equiv 0.5\Delta^k$
$0.1 < \rho^k < 0.75$	Accept step	$\Delta^{k+1} \equiv \Delta^k$
$0.75 \leq \rho^k$	Accept step	$\Delta^{k+1} \equiv 2\Delta^k$

Trust-Region Algorithm for Iteration k

1. Compute a step, $\mathbf{s}_k$, by solving the trust-region subproblem,
$$\min_{\mathbf{s}_k} \quad m_k(\mathbf{x}_k + \mathbf{s}_k)$$
$$\text{s.t.} \quad \|\mathbf{s}_k\| \leq \Delta_k.$$

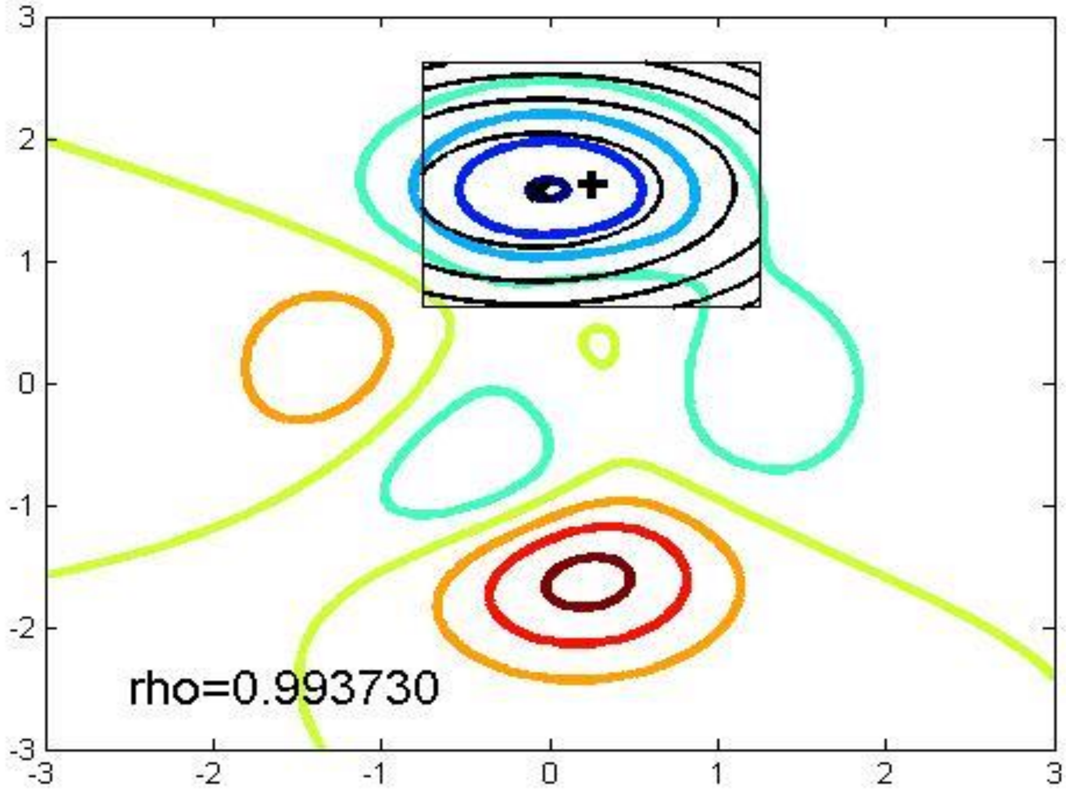
2. Evaluate $f_{\text{high}}(\mathbf{x}_k + \mathbf{s}_k)$.

3. Compute the ratio of actual improvement to predicted improvement,
$$\rho_k = \frac{f_{\text{high}}(\mathbf{x}_k) - f_{\text{high}}(\mathbf{x}_k + \mathbf{s}_k)}{m_k(\mathbf{x}_k) - m_k(\mathbf{x}_k + \mathbf{s}_k)}.$$

4. Accept or reject the trial point according to ρ_k,
$$\mathbf{x}_{k+1} = \begin{cases} \mathbf{x}_k + \mathbf{s}_k & \text{if } \rho_k > 0 \\ \mathbf{x}_k & \text{otherwise.} \end{cases}$$

5. Update the trust region size according to ρ_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: $a(\mathbf{x}) = f_{\text{high}}(\mathbf{x}) - f_{\text{low}}(\mathbf{x})$

  with surrogate constructed as

$$m_k(\mathbf{x}) = f_{\text{low}}(\mathbf{x}) + a(\mathbf{x}_k) + \nabla a(\mathbf{x}_k)^T(\mathbf{x} - \mathbf{x}_k)$$

- Multiplicative correction: $\beta(\mathbf{x}) = \dfrac{f_{\text{high}}(\mathbf{x})}{f_{\text{low}}(\mathbf{x})}$

  with surrogate constructed as

$$m_k(\mathbf{x}) = \left[\beta(\mathbf{x}_k) + \nabla\beta(\mathbf{x}_k)^T(\mathbf{x} - \mathbf{x}_k)\right] f_{\text{low}}(\mathbf{x})$$

- 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]

Trust-region model management: Derivative-free framework

- Derivative-free trust region approaches
  [Conn, Scheinberg, and Vicente, 2009]

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

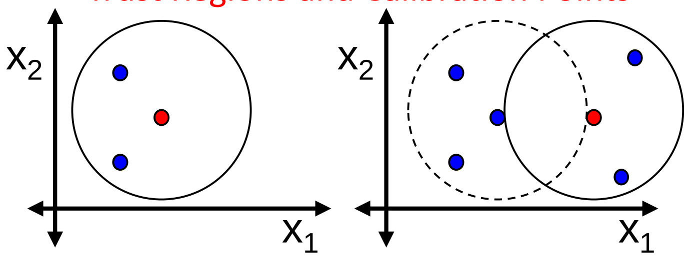
$$\left\| \nabla f_{high}(\mathbf{x}) - \nabla m_k(\mathbf{x}) \right\| \le \kappa_g \Delta_k$$

$$\left| f_{high}(\mathbf{x}) - m_k(\mathbf{x}) \right| \le \kappa_f \Delta_k^2$$

- 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

Trust Regions and Calibration Points



Multifidelity design optimization example: Aircraft wing (with black-box codes)

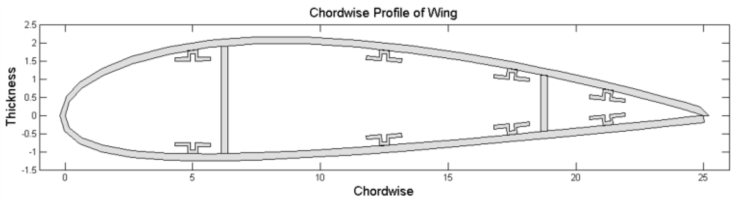
March PhD 2012; March, W., 2012

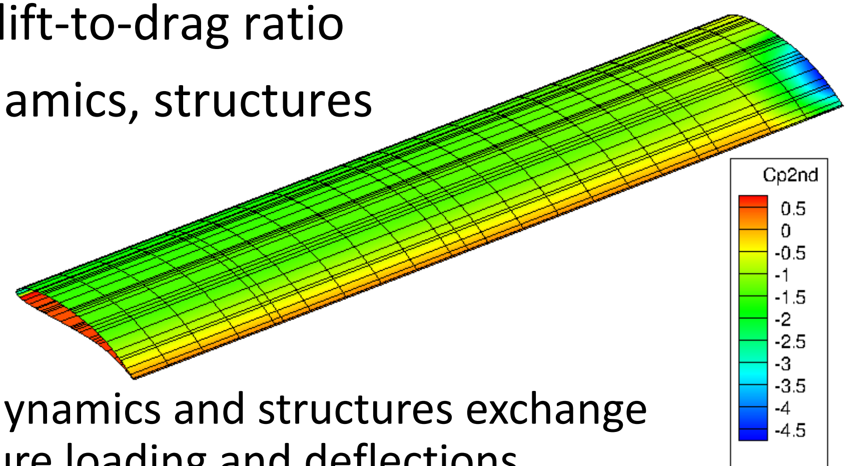
$$\min_{x} f(x)$$
$$\text{s.t.} \quad g(x) \le 0$$
$$h(x) = 0$$

Design variables: wing geometry, structural members

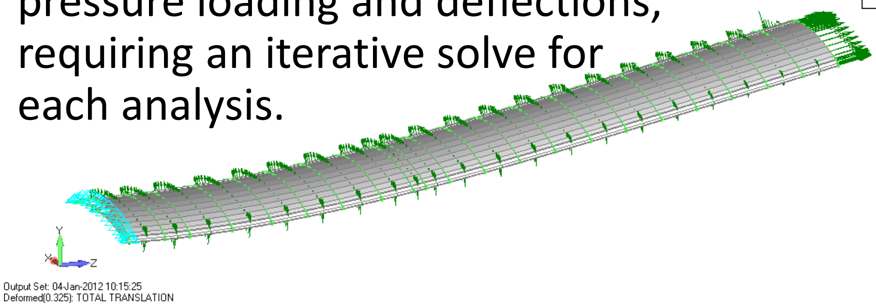
Objectives: weight, lift-to-drag ratio

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

Low-Fidelity Model	Nastran Evals.	Panair Evals.	Time* (days)
None	7,425	7,425	4.73
AVL/Beam Model	5,412	5,412	3.45
Kriging Surrogate	3,232	3,232	2.06

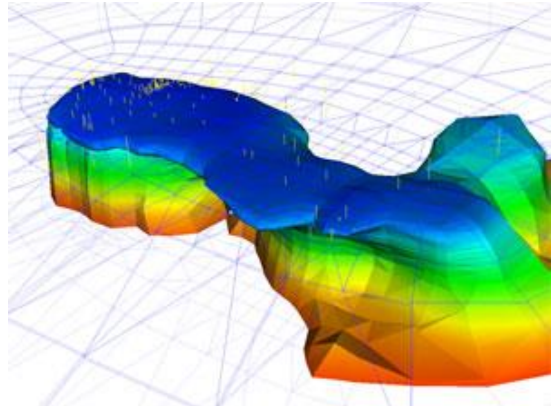
* Time corresponds to average of 30s per Panair evaluation, 25s per Nastran evaluation, and serial analysis of designs within a discipline.

$$\pi(x|d) \sim L(d|x)\, \pi_0(x)$$

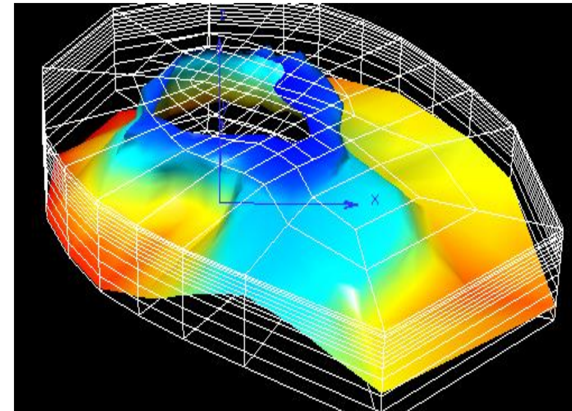
INVERSE PROBLEMS

Large-scale statistical inverse problems

Data



State



Parameters

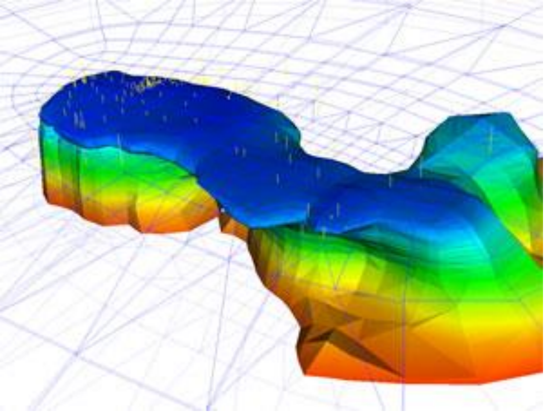
Observation: $d = C(u, e)$

PDE: $\dfrac{\partial u}{\partial t} = A(u, x)$

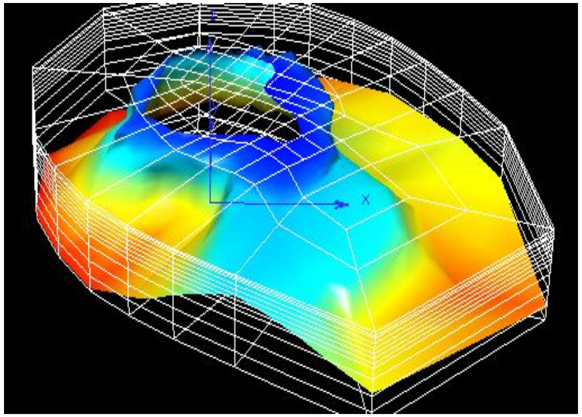
- Data are limited in number, noisy, and indirect
- State-space is high dimensional (PDE model)
- Unknown parameters are high-dimensional

Large-scale statistical inverse problems

Data



State

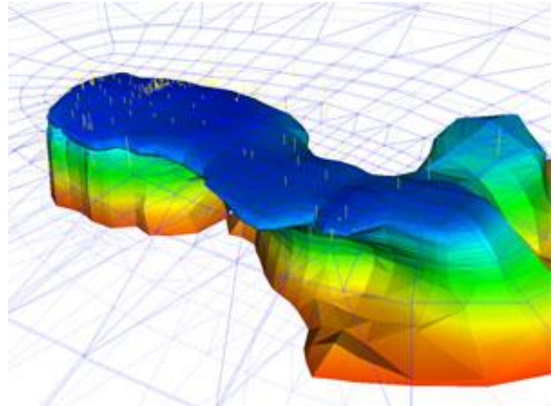


Parameters

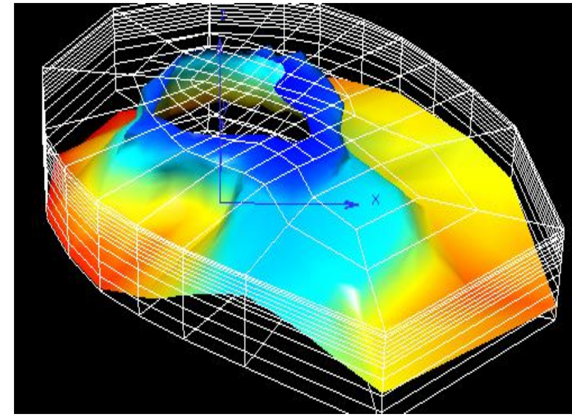
Bayes rule: $\underbrace{\pi(x|d)}_{\text{posterior}} \sim \underbrace{L(d|x)}_{\text{likelihood}} \; \underbrace{\pi_0(x)}_{\text{prior}}$

Large-scale statistical inverse problems: Exploiting low-rank structure

Data



State



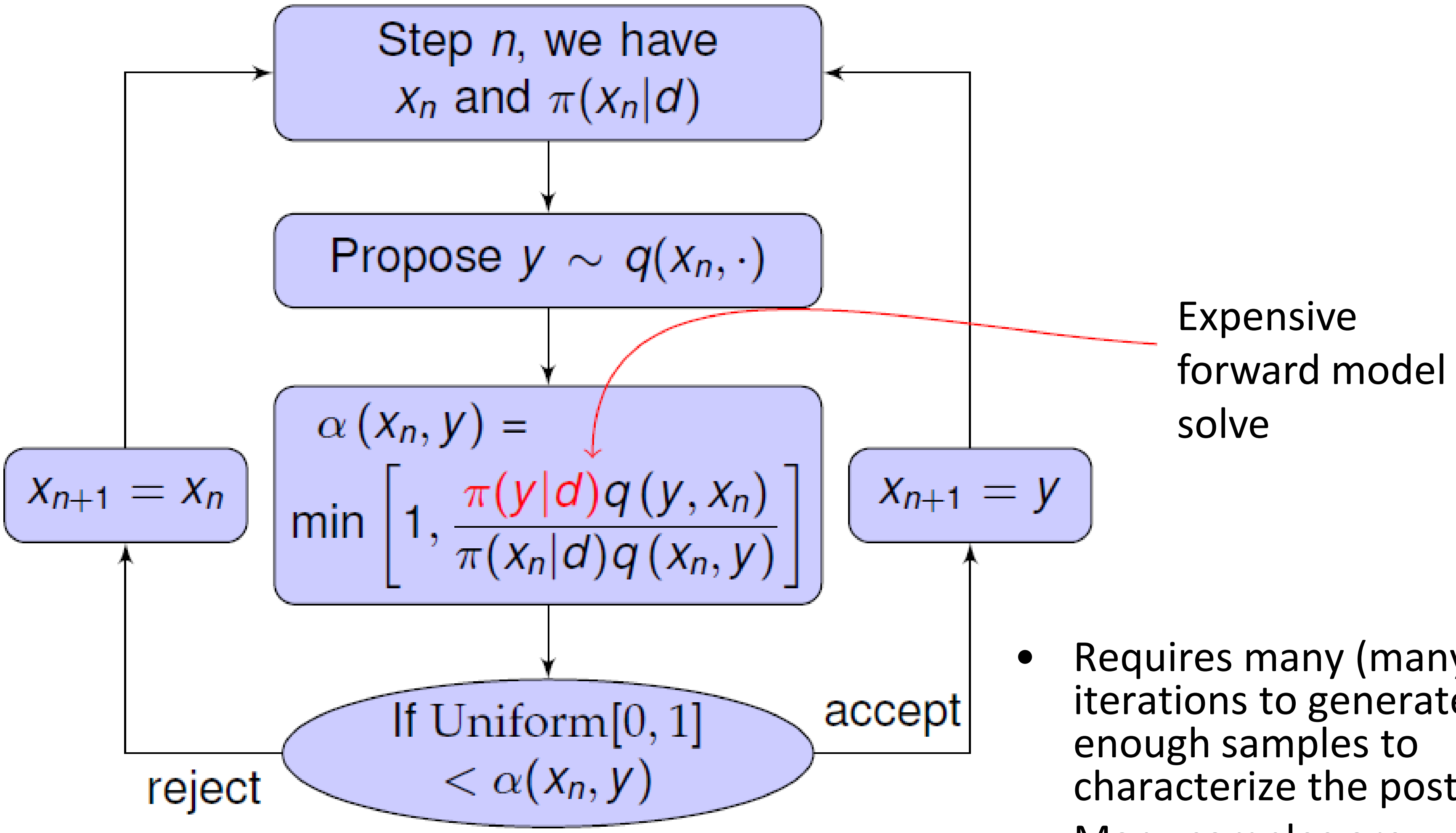
Parameters

Bayes rule: $$\underbrace{\pi(x|d)}_{\text{posterior}} \sim \underbrace{L(d|x)}_{\text{likelihood}} \; \underbrace{\pi_0(x)}_{\text{prior}}$$

- 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]

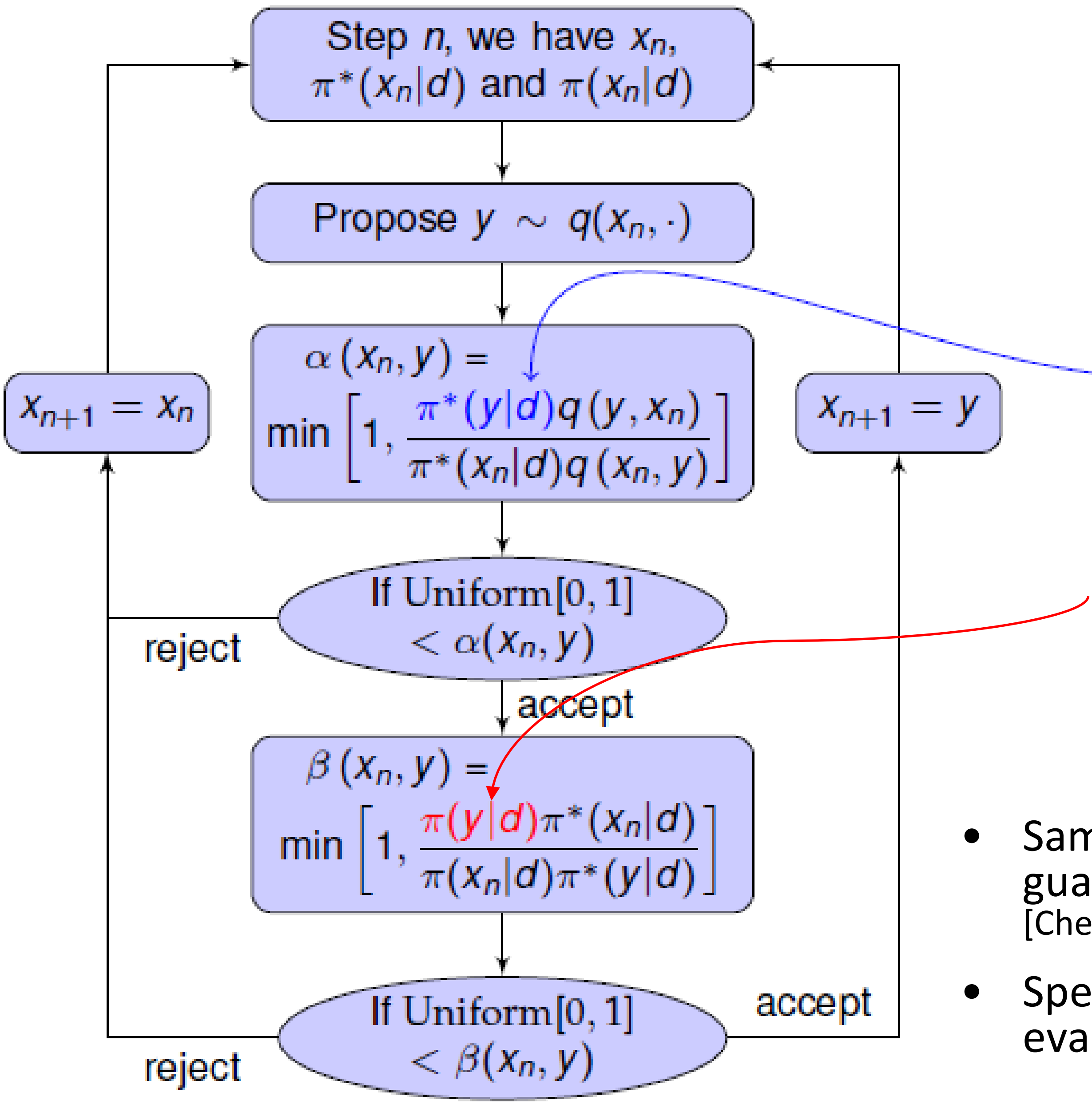
Exploring the posterior: MCMC Sampling

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

Multifidelity: Adaptive delayed acceptance MCMC sampling



Approximate (cheap) forward model solve

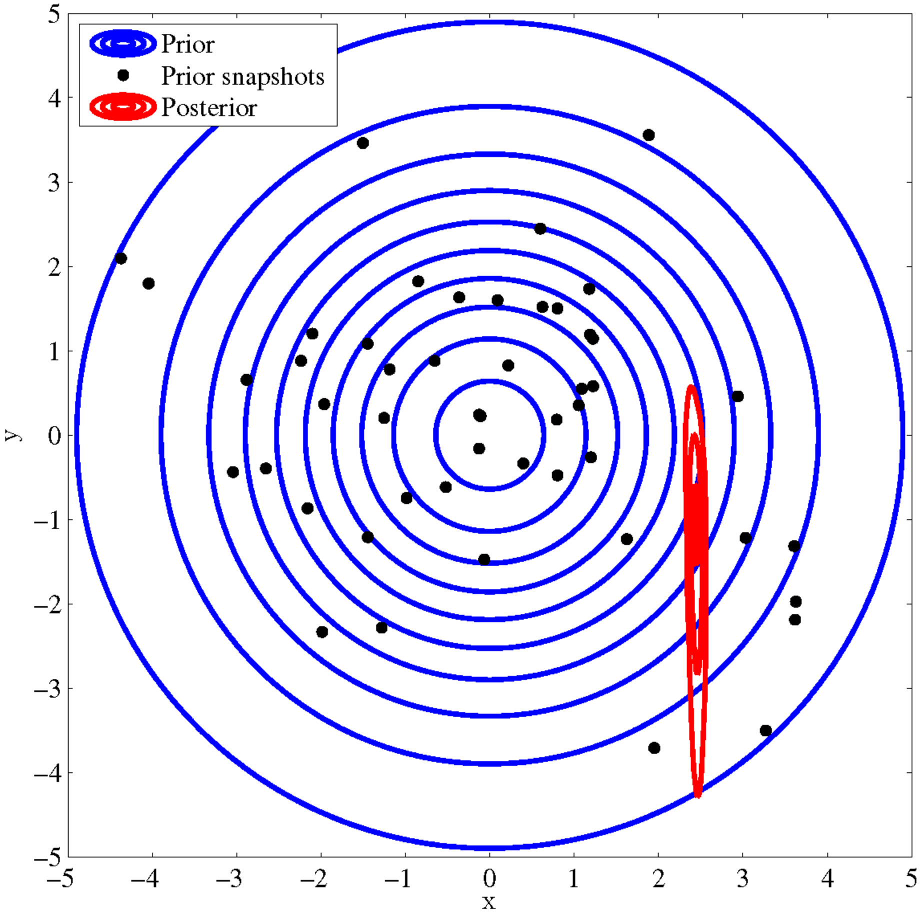
Full model evaluation ensures sampling the exact posterior

- 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

Cui, Marzouk, W., 2014

- 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

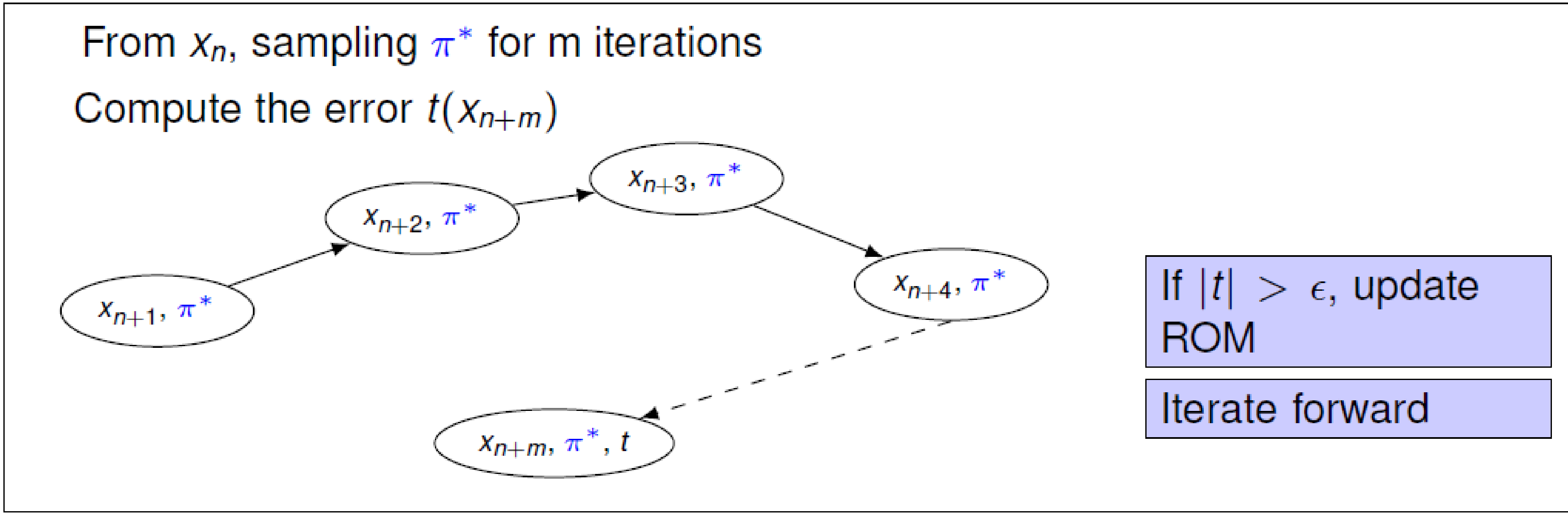
Simultaneous model reduction and posterior exploration

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

- Stage 1:
  - At each MCMC iteration, first sample the approximate posterior distribution (π^*) 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]

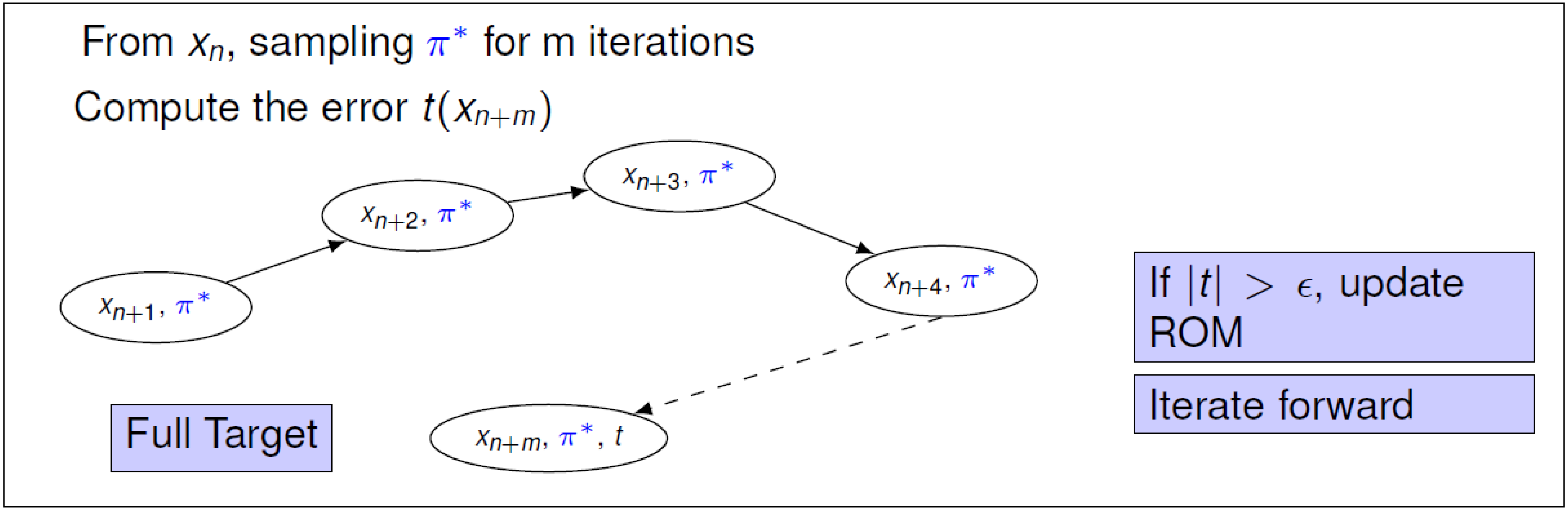
- Stage 2:
  - The last state of the Stage 1 Markov chain is the proposal candidate
  - Compute acceptance probability (α) based on full posterior density value (ensures that we sample the exact posterior)
  - After each full posterior density evaluation, the state of the associated forward model evaluation is a potential new snapshot

Simultaneous model reduction and posterior exploration

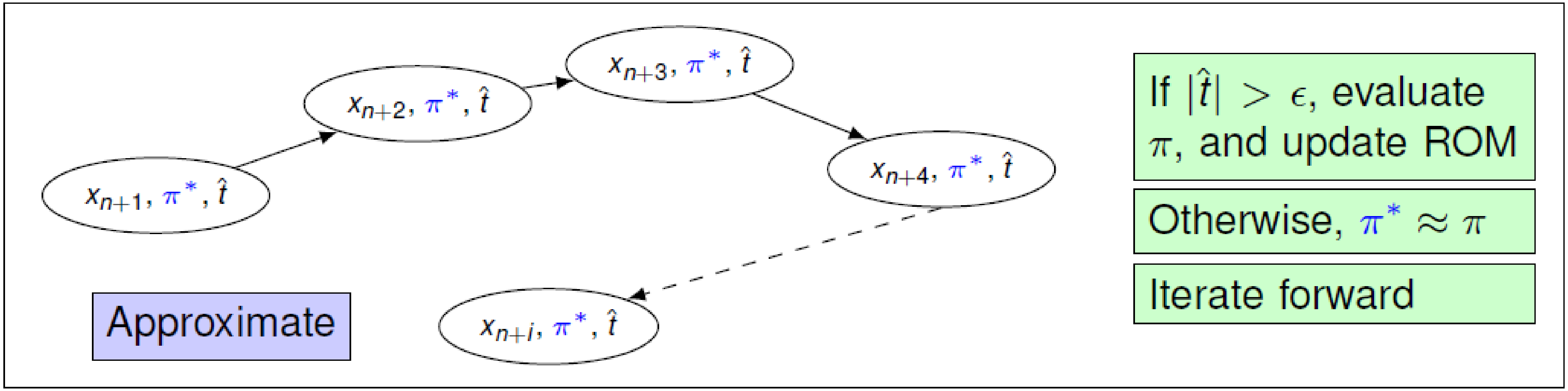


- Compute the error of the reduced model output estimate at each new posterior sample
- Update the reduced basis with the new snapshot when the error exceeds a threshold ϵ
- The resulting reduced model is data-driven, since it uses the information provided by the observed data (in the form of the posterior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

Simultaneous model reduction and posterior exploration



- 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



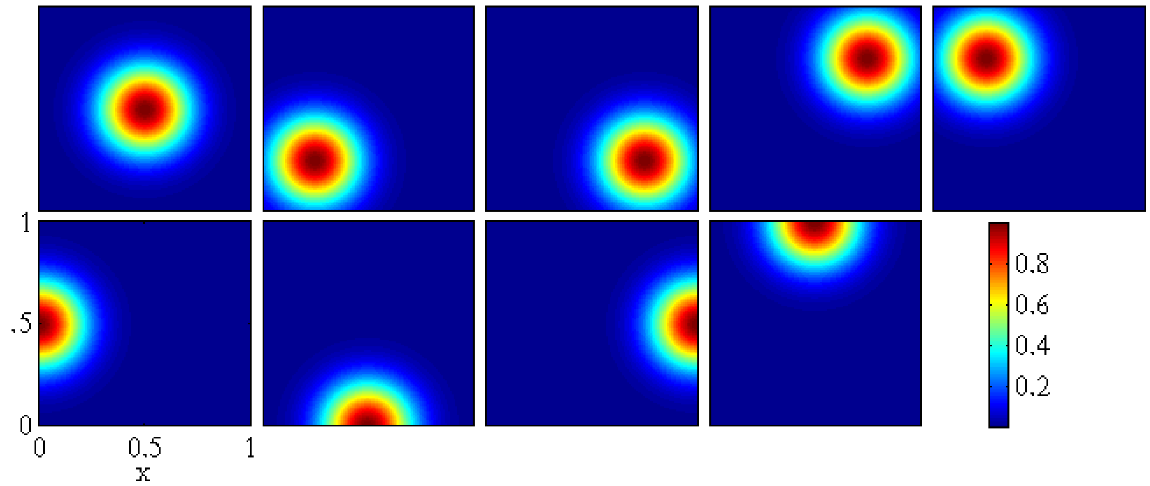
Inverse problem example: 9D test case

$$
\begin{aligned}
-\nabla \cdot (k(r)\nabla u(r)) &= f(r), \;\; r \in D \\
k(r)\nabla u(r) \cdot \vec{n}(r) &= 0, \qquad r \in \partial D
\end{aligned}
$$

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

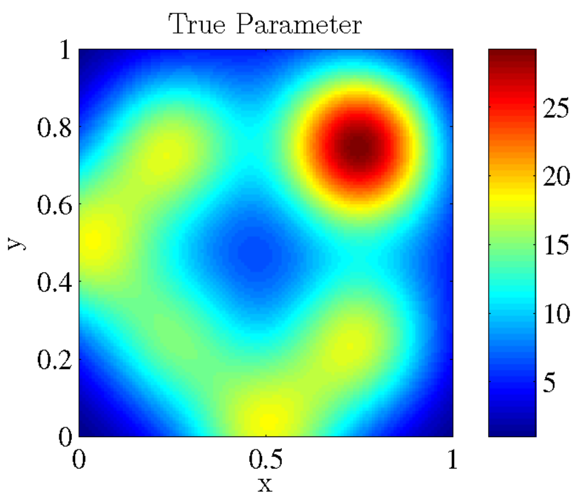
$$k(r) = \sum_{i=1}^{9} b_i(r) x_i$$

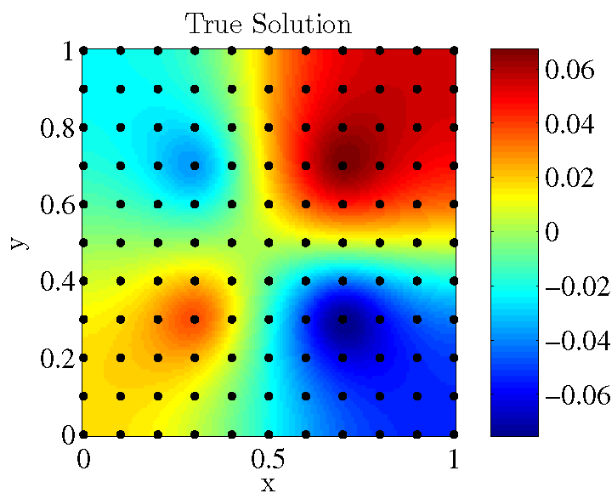
$$\log(x_i) \sim \mathcal{N}(\mu_i, \sigma_i^2)$$



121 potential measurements, signal to noise ratio 50.

Full model has 120×120 elements.



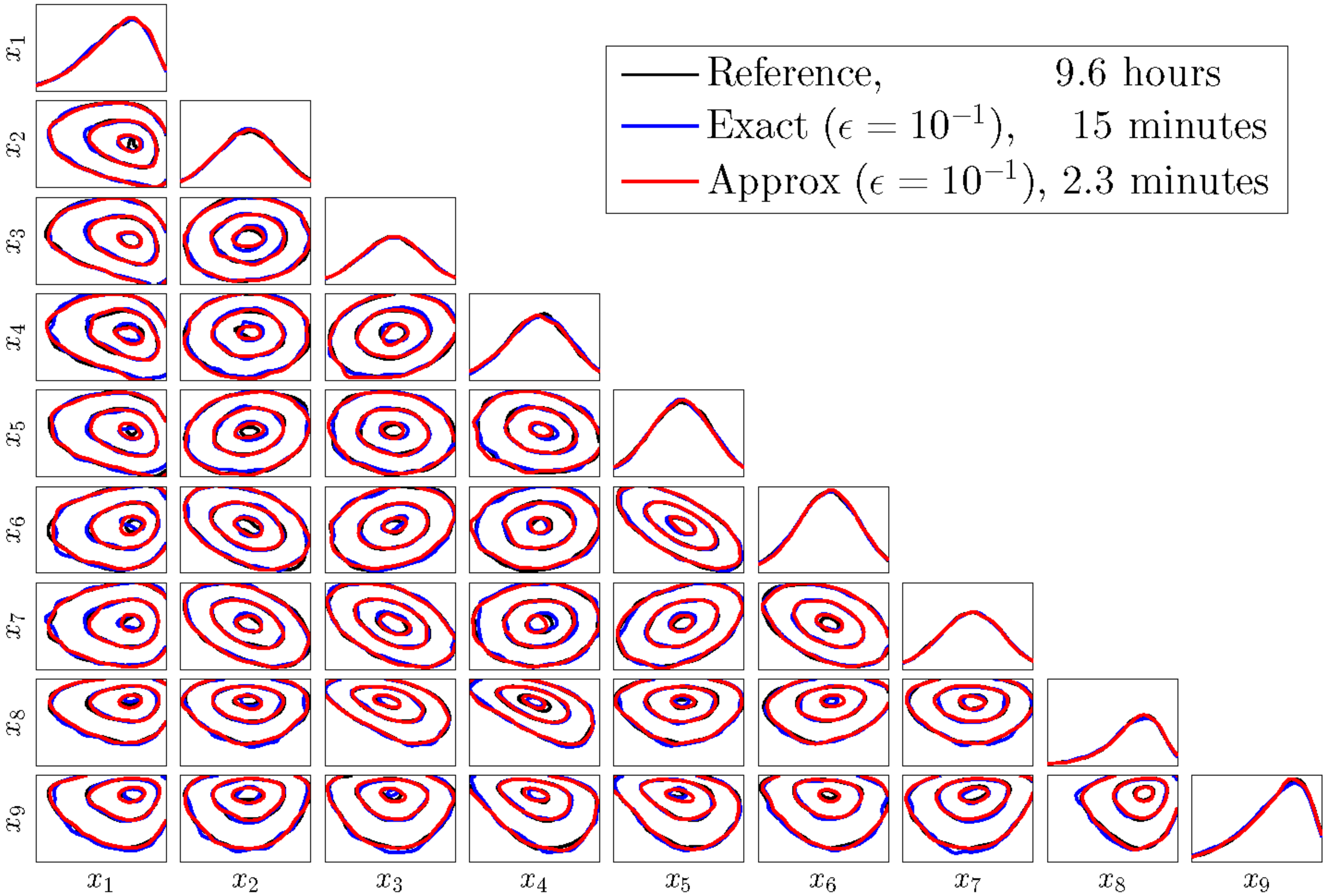


Inverse problem example: Sampling efficiency

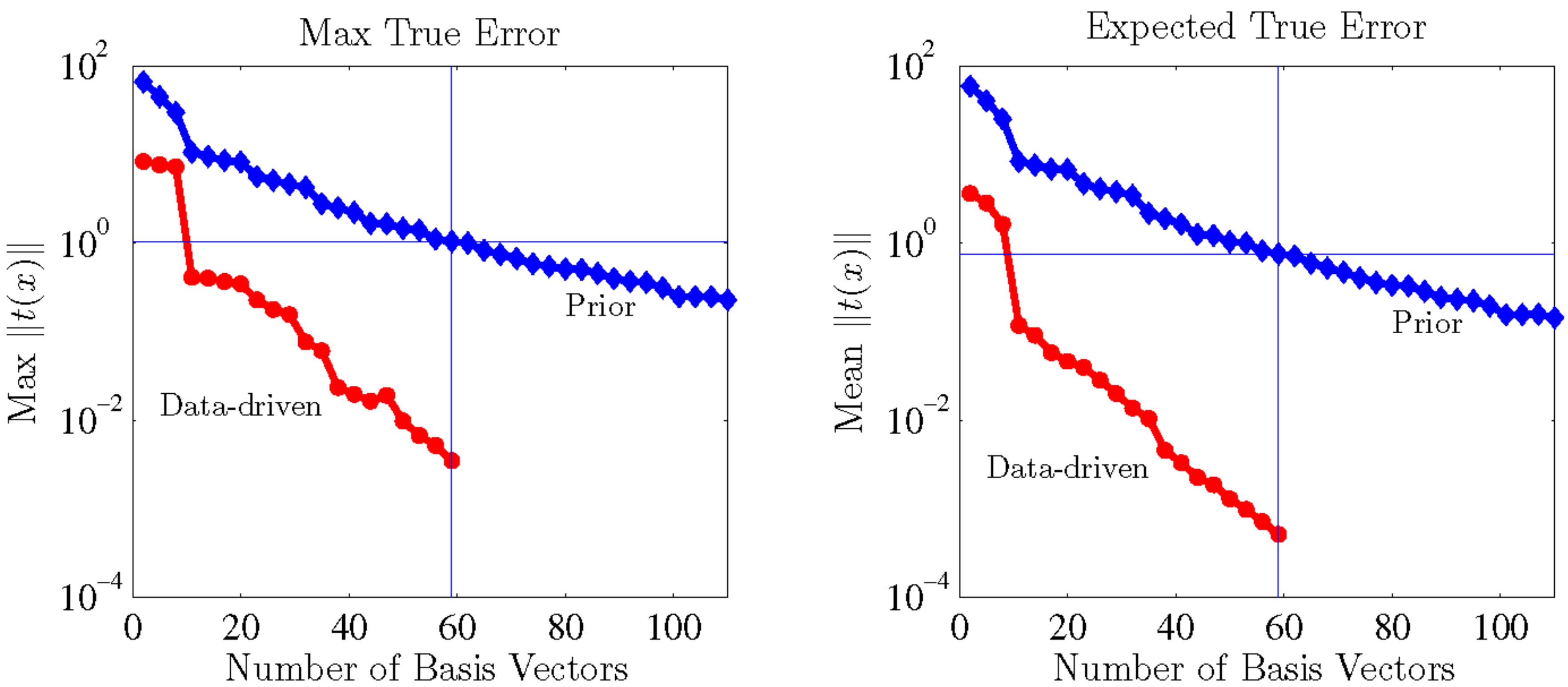
	Reference	Full Target			Approximate		
Error threshold ϵ	-	10^{-1}	10^{-2}	10^{-3}	10^{-1}	10^{-2}	10^{-3}
Basis vectors	-	14	33	57	17	35	57
ESS / CPU time	0.058	2.5	2.7	2.6	15	12	8.9
Speed-up factor	1	43	46	45	256	213	154

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

Inverse problem example: Sampling accuracy

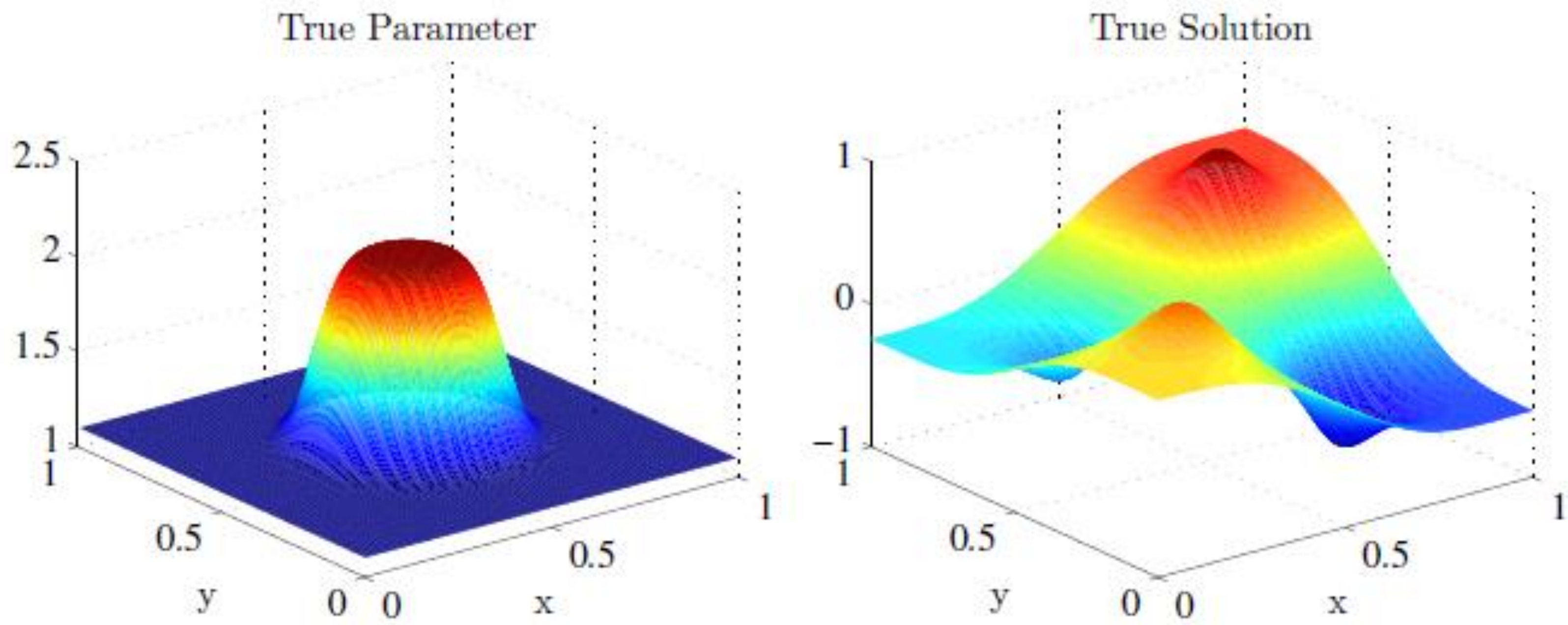


Inverse problem example: Reduced model performance



- 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 $k(r) = \exp[x(r)]$ is spatially distributed, has the prior

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

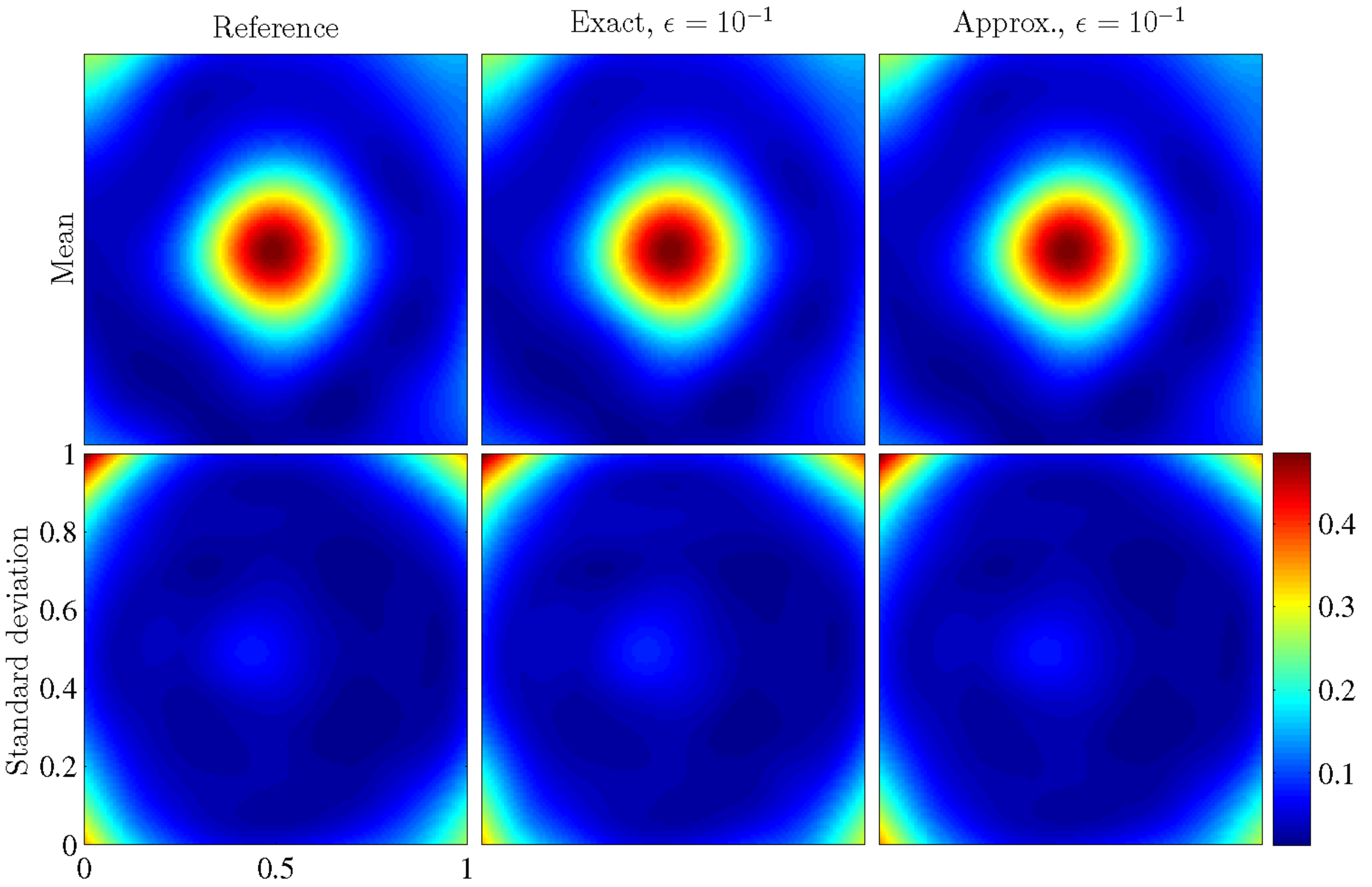
We use $L = 0.25$, and apply Karhunen-Loéve expansion to $\mathcal{C}$. We keep the first 45 modes $\sim$ 99.99% energy.

Inverse problem example: Sampling efficiency

	Reference	Full Target	Approximate		
Error threshold ϵ	-	10^{-1}	10^{-1}	10^{-2}	10^{-3}
Basis vectors	-	64	62	129	209
ESS / CPU time	0.033	2.2	8.2	3.7	2
Speed-up factor	1	67	249	111	61

- Run both algorithms for 5×10^5 iterations.
- A reference MCMC (only based on the full model) is simulated for 5×10^5 iterations.
- Speed-up factor is estimated from CPU time per effective sample.

Inverse problem example: Sampling accuracy



21 hours | 17 minutes | 5 minutes

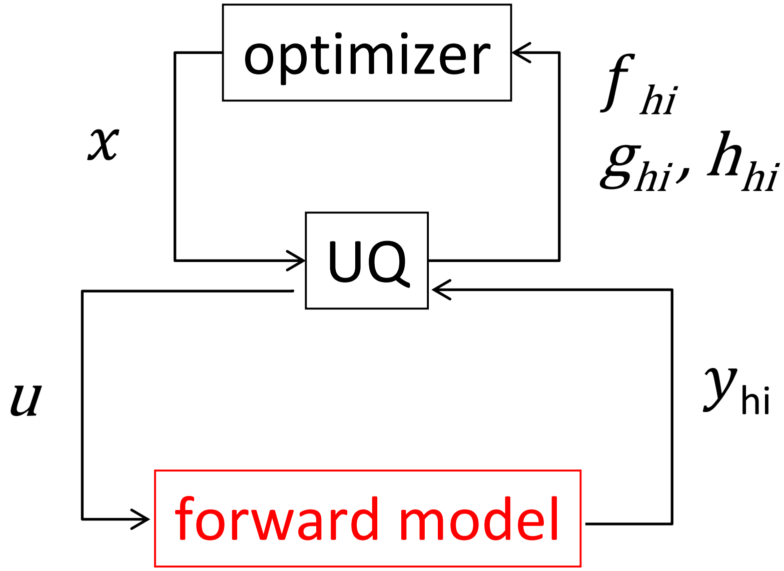
$$
\begin{aligned}
\min_x \ & f\big(x, s(x)\big) \\
\text{s.t.} \ & g\big(x, s(x)\big) \leq 0 \\
& h\big(x, s(x)\big) = 0
\end{aligned}
$$

UNCERTAINTY QUANTIFICATION

The challenge of optimization under uncertainty (OUU)

$$\min_{x} f(x, s(x))$$
$$\text{s.t.} \quad g(x, s(x)) \leq 0$$
$$h(x, s(x)) = 0$$

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



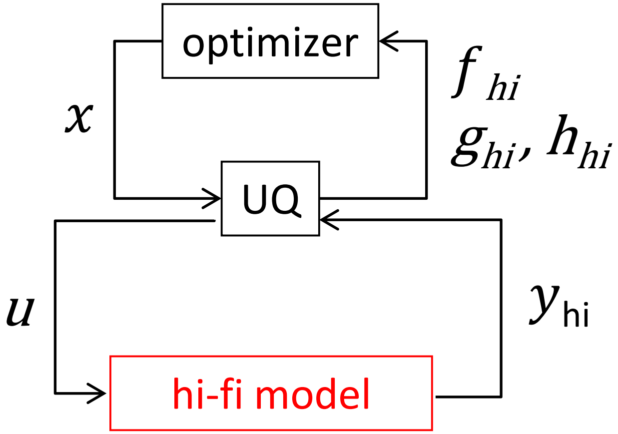
High-fidelity model embedded in a UQ loop in an optimization loop

- Large computational cost
- Need an optimizer that is tolerant to noisy estimates of statistics

Multifidelity optimization under uncertainty

$$\min_x f\big(x, s(x)\big)$$
$$\text{s.t.} \quad g\big(x, s(x)\big) \leq 0$$
$$h\big(x, s(x)\big) = 0$$

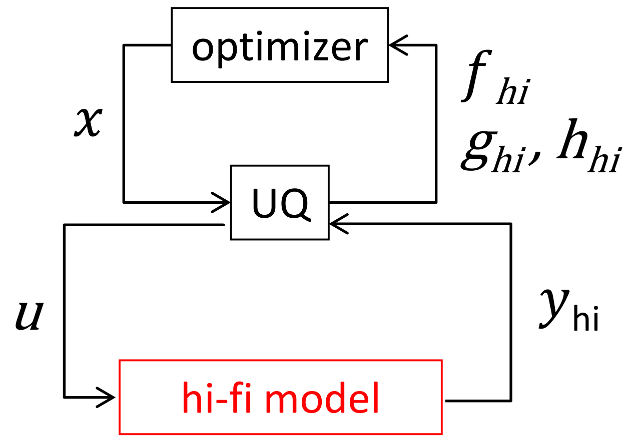
Design variables	x
Uncertain parameters	u
Model outputs	$y(x,u)$
Statistics of model	$s(x)$

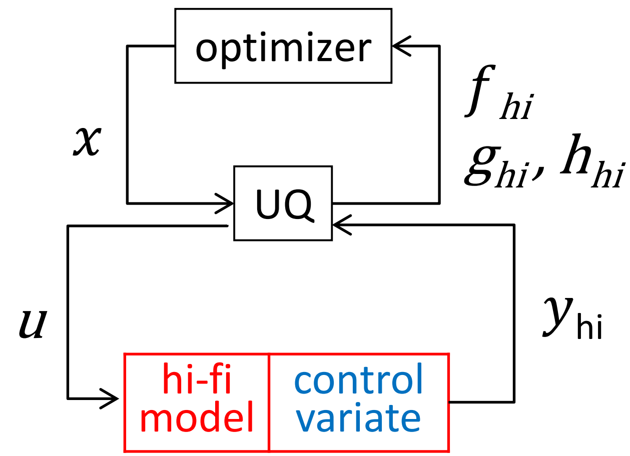


Multifidelity OUU approach: Control variates

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

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

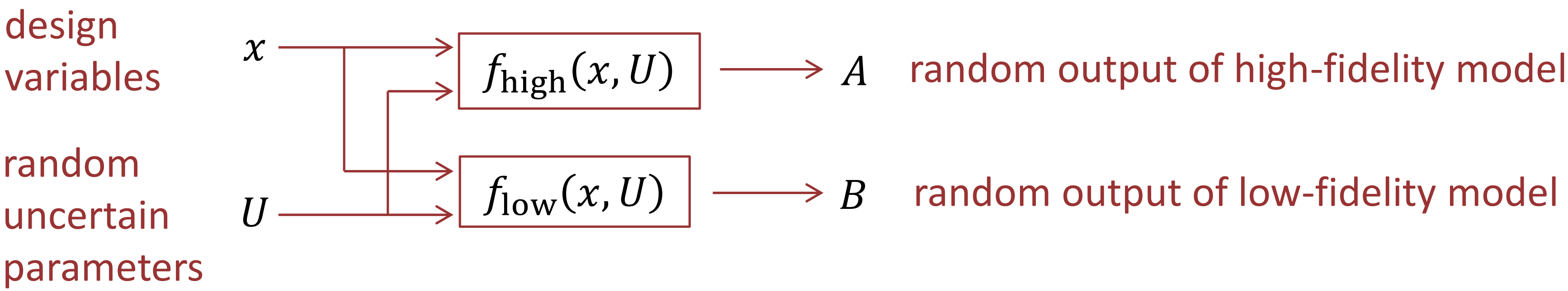




Control variates: Exploit model correlation

- 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



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

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

$$
\begin{aligned}
&\min_x f\big(x, s_A(x)\big) \\
&\text{s.t.} \quad g\big(x, s_A(x)\big) \le 0
\end{aligned}
$$

approximated by

$$
\begin{aligned}
&\min_x f(x, \hat{s}_A) \\
&\text{s.t.} \quad g\big(x, \hat{s}_A(x)\big) \le 0
\end{aligned}
$$

Variance reduction with control variate

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

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

Definitions:

$$\sigma_A^2 = \mathrm{Var}[A]$$

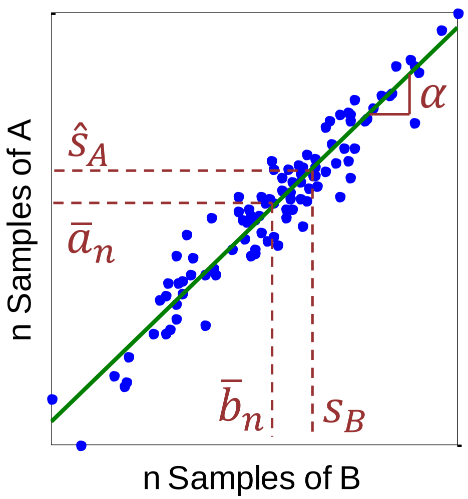
$$\sigma_B^2 = \mathrm{Var}[B]$$

$$\rho_{AB} = \mathrm{Corr}[A, B]$$

- Control variate (CV) estimator of s_A:
  - Additional random variable B with known $s_B = \mathbb{E}[B]$

$$\hat{s}_A = \bar{a}_n + \alpha\left(s_B - \bar{b}_n\right)$$

$$\mathrm{Var}[\hat{s}_A] = \frac{\sigma_A^2 + \alpha^2\sigma_B^2 - 2\alpha\rho_{AB}\sigma_A\sigma_B}{n}$$



- Minimize $\mathrm{Var}[\hat{s}_A]$ with respect to α

$$\mathrm{Var}[\hat{s}_A^*] = \underbrace{(1 - \rho_{AB}^2)}_{\leq 1}\frac{\sigma_A^2}{n}$$

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)

Definitions:

$$\sigma_A^2 = \mathrm{Var}[A]$$

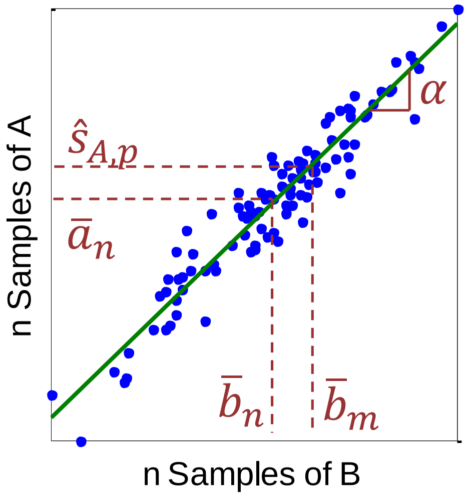
$$\sigma_B^2 = \mathrm{Var}[B]$$

$$\rho_{AB} = \mathrm{Corr}[A, B]$$

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

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

- Using difference $\left(\bar{b}_m - \bar{b}_n\right)$ as correction to $\bar{a}_n$
- Leveraging correlation between A and B
  - Correlation captured in α



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 $\mathrm{Var}\left[\hat{s}_{A,p}\right]$ with respect to α and r

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

- Limiting cases:

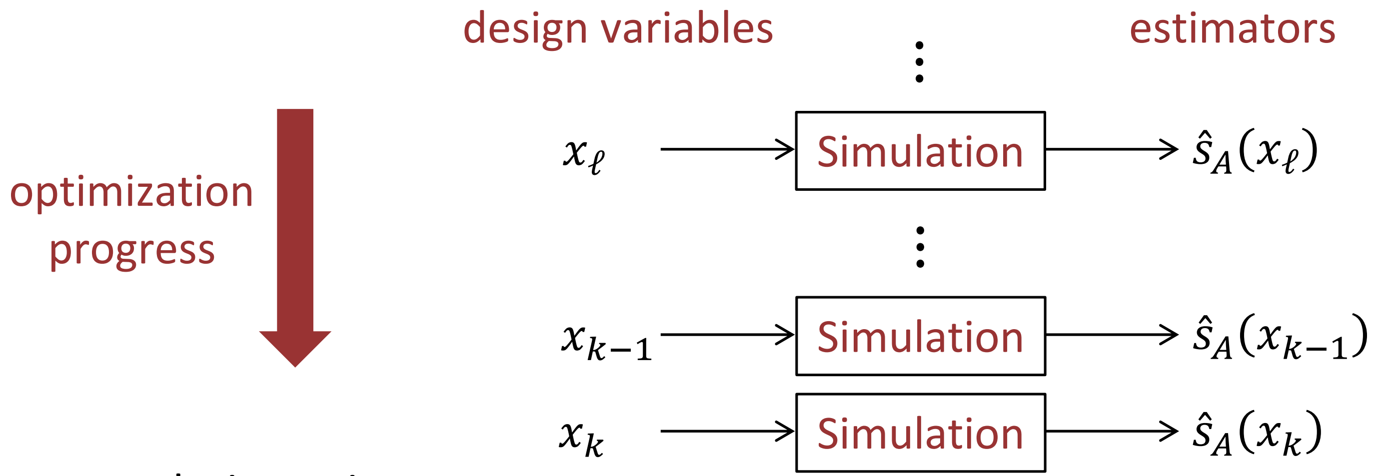
(i) Low-fidelity model “free”: as $w \to \infty$, then $\mathrm{Var}\left[\hat{s}_{A,p}^*\right] \to (1 - \rho_{AB}^2)\frac{\sigma_A^2}{p}$

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

Definitions: $\sigma_A^2 = \mathrm{Var}[A]$, $\sigma_B^2 = \mathrm{Var}[B]$, $\rho_{AB} = \mathrm{Corr}[A, B]$

Model correlation over design space

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



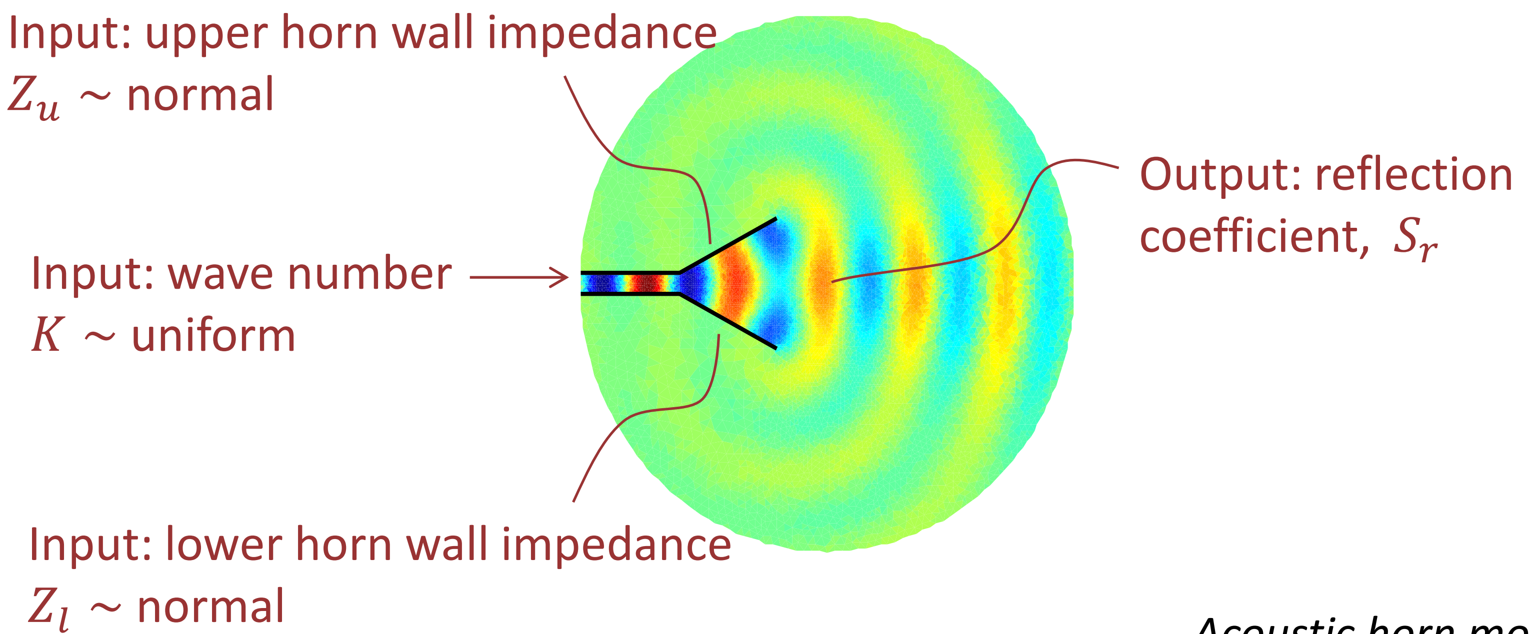
- At current design point x_k
  - Define $A = M_{\text{high}}(x_k, U)$
  - Want to compute $\hat{s}_A$ as estimator of $s_A = \mathbb{E}[A]$

Information Reuse Estimator

- Previously visited design point x_ℓ 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 $\text{Var}[\hat{s}_C]$

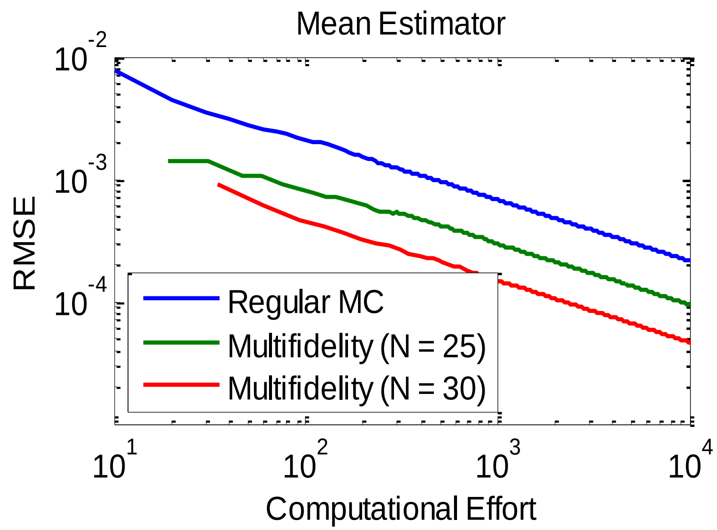
Acoustic horn example

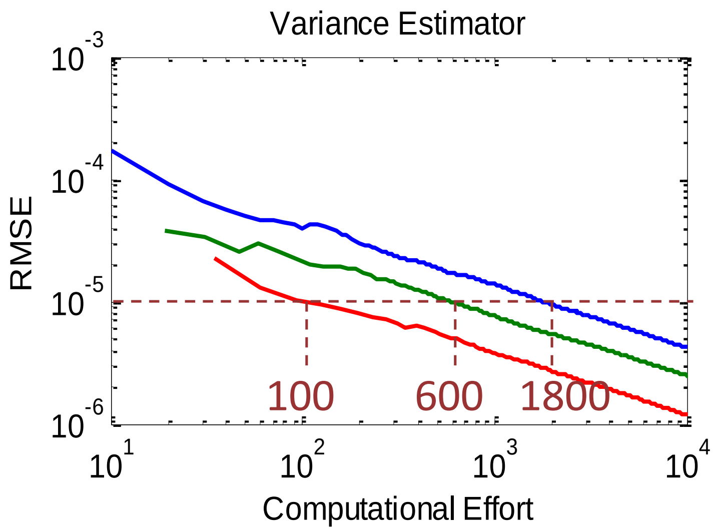
- Helmholtz equation for propagation of acoustic waves through 2-D horn
  - High-fidelity model: Finite element model (FEM) with 35,895 states
  - Low-fidelity model I: Reduced basis model (RBM) with N = 25 states
  - Low-fidelity model II: Reduced basis model (RBM) with N = 30 states
  - Ratio of evaluation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

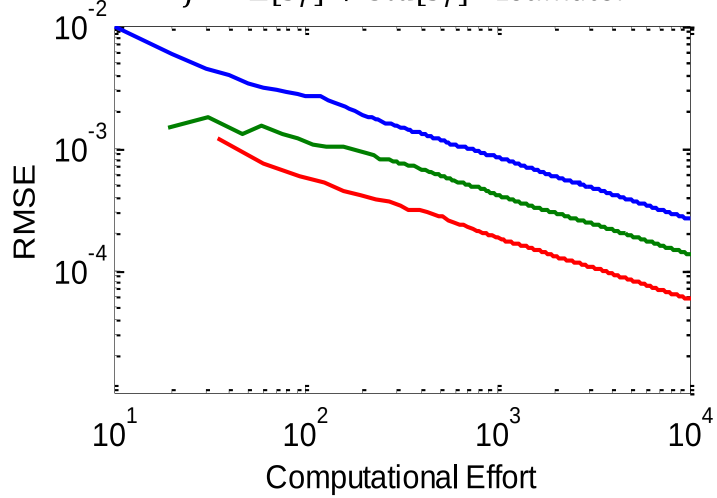


Acoustic horn models due to D.B.P. Huynh

Acoustic horn example – uncertainty propagation

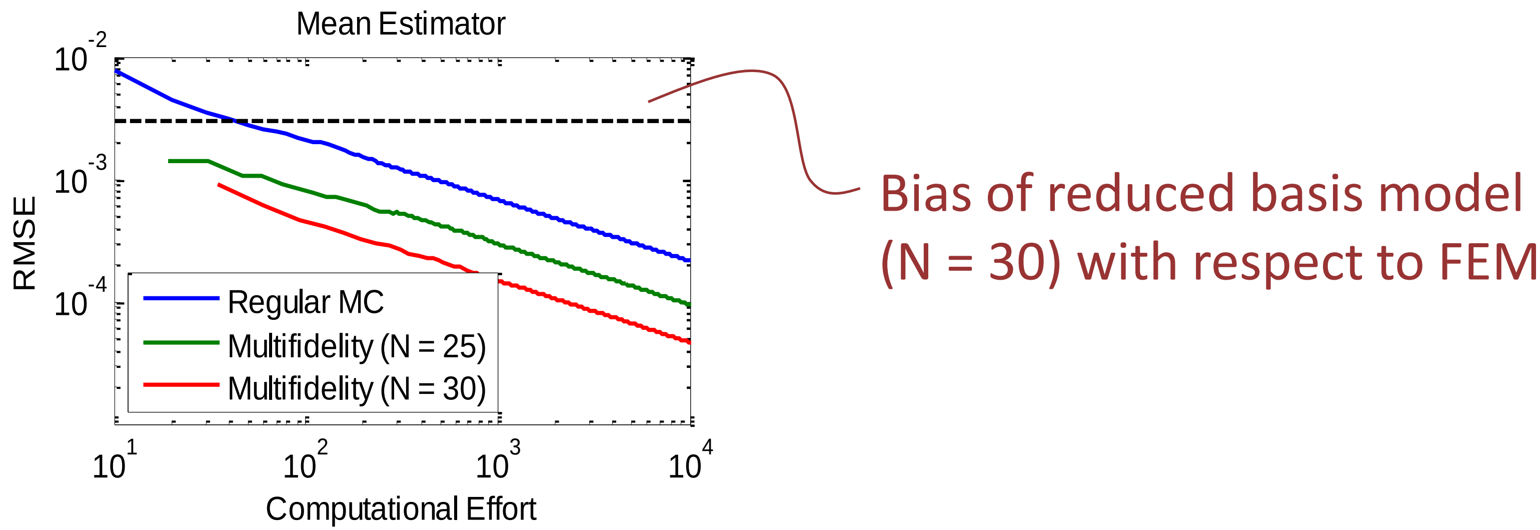






- $w = 40$ in both cases
- Correlation between FEM and
  - RBM (N = 25) ≈ 0.928
  - RBM (N = 30) ≈ 0.996
- Increasing correlation increases efficiency of multifidelity estimator

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

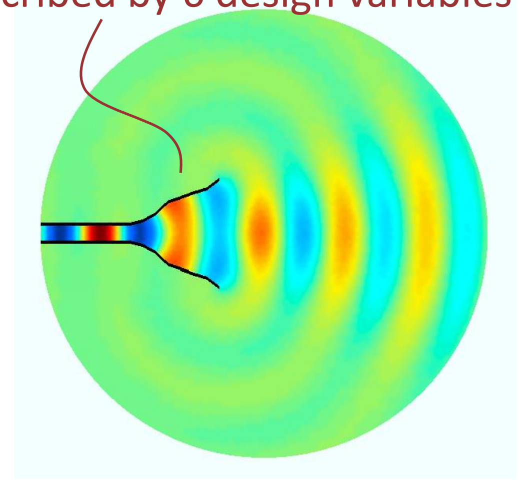
Decision variables: horn geometry, *b*

Uncertainty: wavenumber, wall impedances

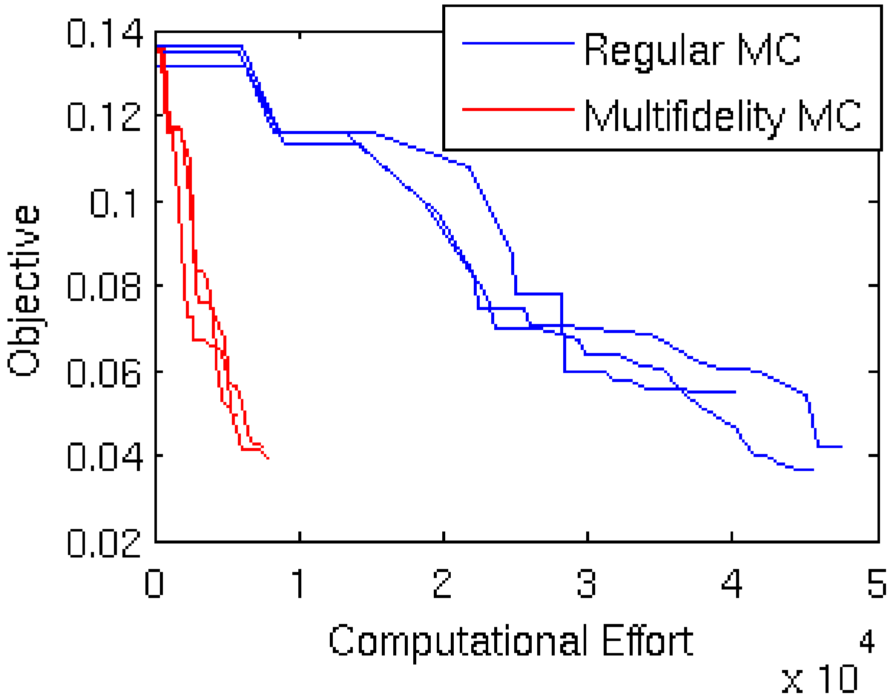
Output of interest: reflection coefficient, s_r

$$\min_b \mathbb{E}[s_r] + \sqrt{\mathbb{V}\mathrm{ar}[s_r]}$$

Robust optimal horn flare shape described by 6 design variables



	Equivalent number of hi-fi evaluations	
Regular MC	44,343	
Multifidelity MC	6,979	(-84%)



Optimization algorithm:

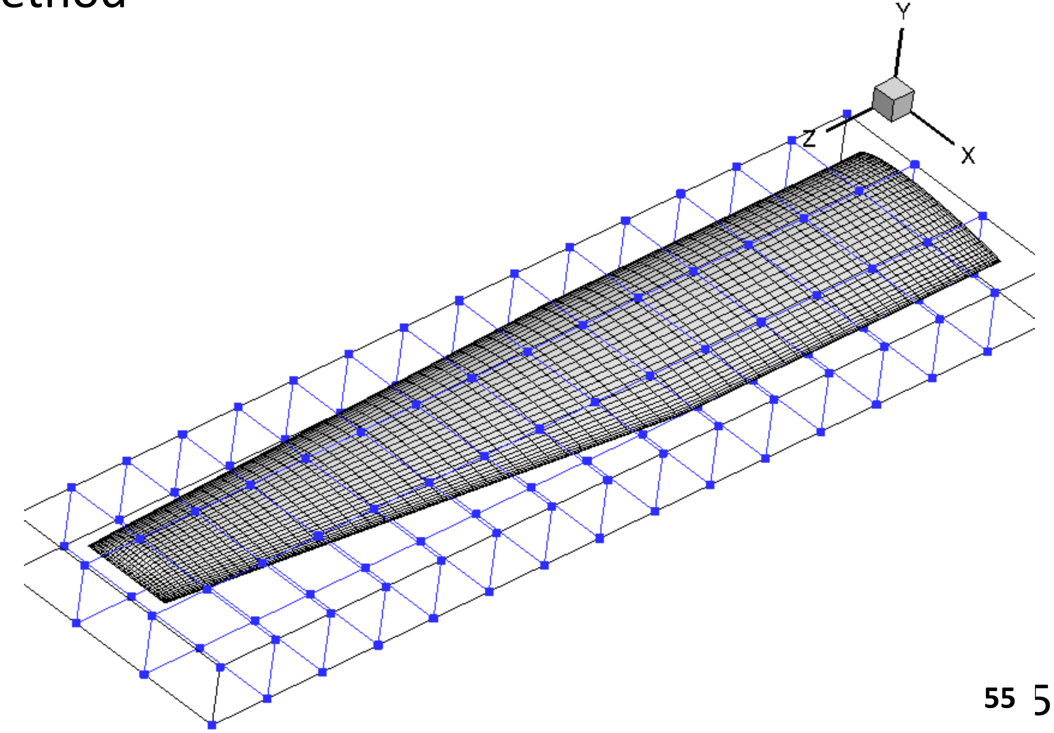
Implicit filtering [Kelley, 2011]

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

	Coarse	Fine
Aerodynamic Panels	1000	2960
Structural d.o.f.	5624	14,288
Eval time	6 s	24 s

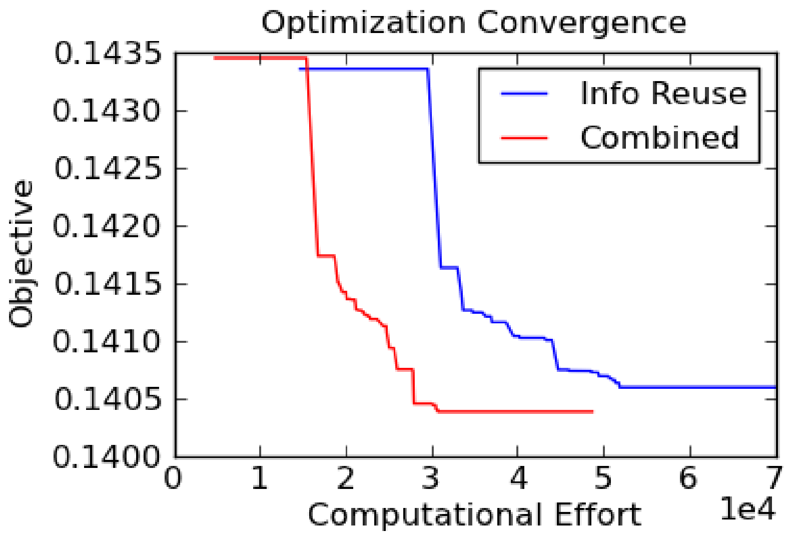


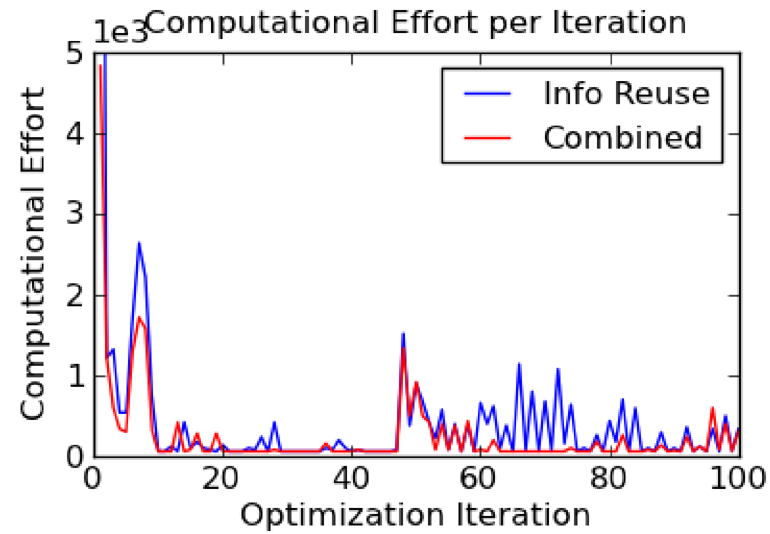
High-fidelity wing optimization

- 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

56

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

	Computational Effort	Total Time (days)
Regular MC	--	--
Info Reuse	7×10^4	13.4
Combined	5×10^4	9.7

Summary

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}
- 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