An acceleration framework for parameter estimation using implicit sampling and adaptive reduced-order models

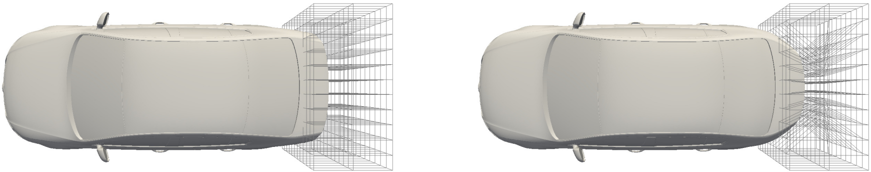
Robert Baraldi, Matthias Morzfeld, Matthew J. Zahr[†]

MS91: Large-scale PDE-constrained optimization algorithms and applications
SIAM Conference on Computational Science and Engineering
Spokane Convention Center, Spokane, Washington, USA
Febrary 25 - March 1, 2019

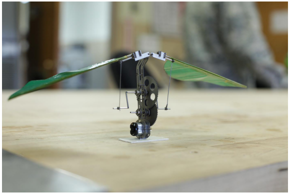
[†] Department of Aerospace and Mechanical Engineering
University of Notre Dame

PDE optimization is ubiquitous in science and engineering

Design: Find system that optimizes performance metric, satisfies constraints



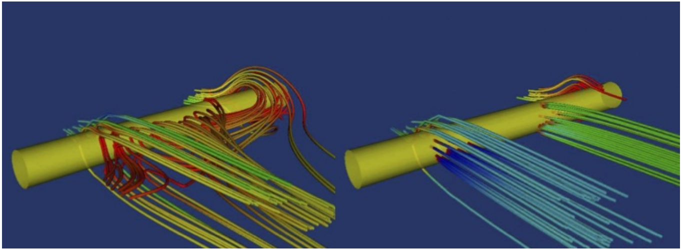
Aerodynamic shape design of automobile



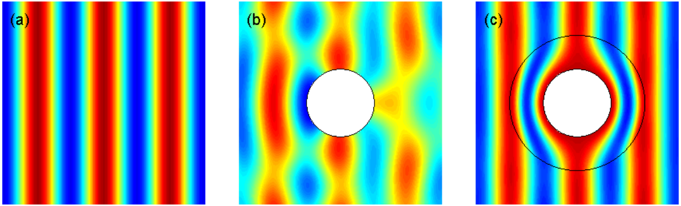
Optimal flapping motion of micro aerial vehicle

PDE optimization is ubiquitous in science and engineering

Control: Drive system to a desired state



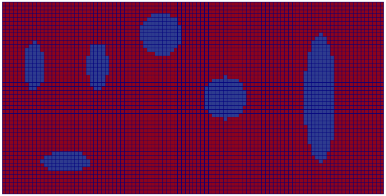
Boundary flow control

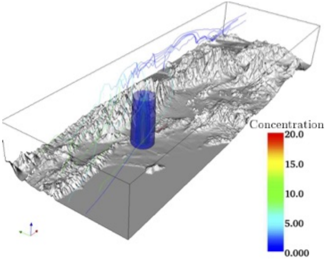


Metamaterial cloaking – electromagnetic invisibility

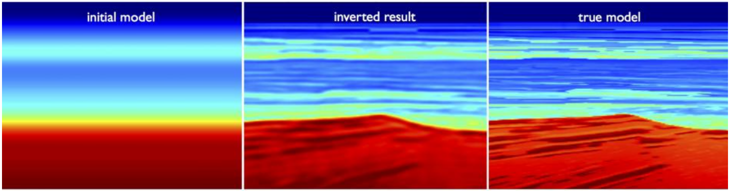
PDE optimization is ubiquitous in science and engineering

Inverse problems: Infer the problem setup given solution observations





Left: Material inversion: find defects from acoustic, structural measurements
Right: Source inversion: find source of airborne contaminant from measurements



Full waveform inversion: estimate subsurface from acoustic measurements

Deterministic[1] PDE-constrained optimization formulation

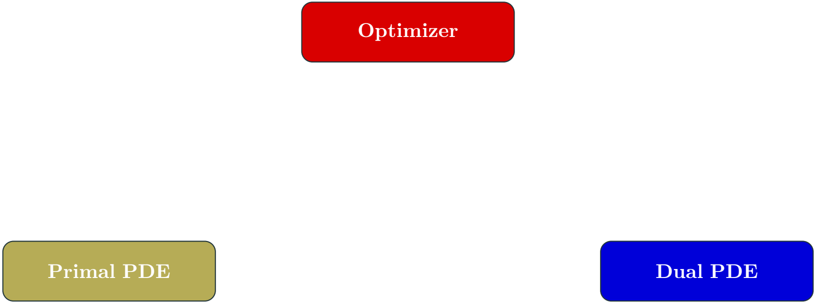
$$\begin{aligned} \underset{\boldsymbol{\mu}\in\mathbb{R}^{n_{\boldsymbol{\mu}}}}{\text{minimize}} \quad & \mathcal{J}(\boldsymbol{u},\, \boldsymbol{\mu}) \\ \text{subject to} \quad & \boldsymbol{r}(\boldsymbol{u},\, \boldsymbol{\mu}) = 0 \end{aligned}$$

$\boldsymbol{r} : \mathbb{R}^{n_{\boldsymbol{u}}} \times \mathbb{R}^{n_{\boldsymbol{\mu}}} \to \mathbb{R}^{n_{\boldsymbol{u}}}$	discretized PDE
$\mathcal{J} : \mathbb{R}^{n_{\boldsymbol{u}}} \times \mathbb{R}^{n_{\boldsymbol{\mu}}} \to \mathbb{R}$	quantity of interest
$\boldsymbol{u} \in \mathbb{R}^{n_{\boldsymbol{u}}}$	PDE state vector
$\boldsymbol{\mu} \in \mathbb{R}^{n_{\boldsymbol{\mu}}}$	optimization parameters

[1] Extension to stochastic see MS280 on Thursday

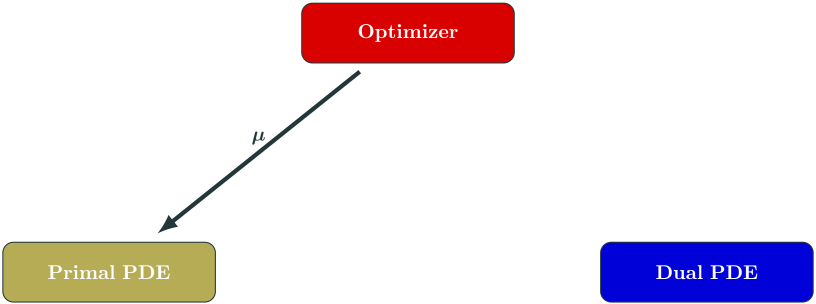
Nested approach to PDE-constrained optimization

Virtually all expense emanates from primal/dual PDE solves



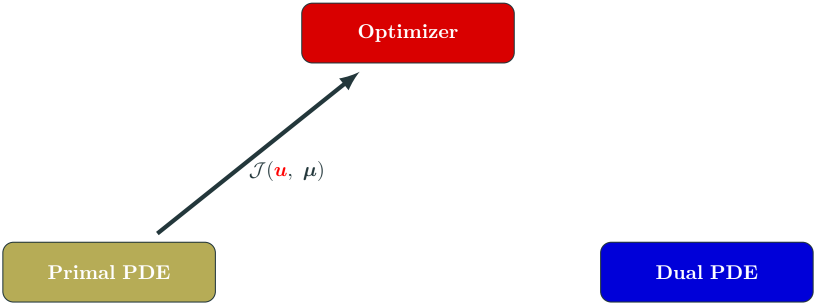
Nested approach to PDE-constrained optimization

Virtually all expense emanates from primal/dual PDE solves



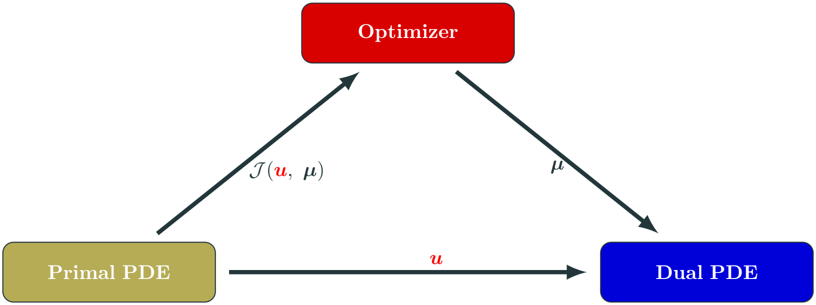
Nested approach to PDE-constrained optimization

Virtually all expense emanates from primal/dual PDE solves



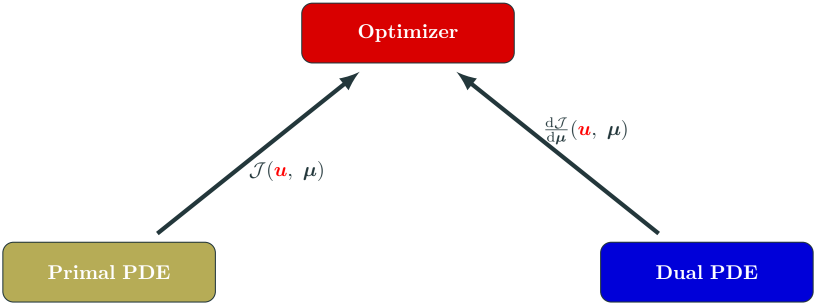
Nested approach to PDE-constrained optimization

Virtually all expense emanates from primal/dual PDE solves



Nested approach to PDE-constrained optimization

Virtually all expense emanates from primal/dual PDE solves



Efficient PDE-constrained optimization using managed inexactness

Application to Bayesian parameter estimation

Efficient PDE-constrained optimization using managed inexactness

Proposed approach: managed inexactness

Replace expensive PDE with inexpensive approximation model

- **Reduced-order models** used for *inexact PDE evaluations*
- **Partially converged solutions** used for *inexact PDE evaluations*

$$\underset{\boldsymbol{\mu} \in \mathbb{R}^{n_{\boldsymbol{\mu}}}}{\text{minimize}} \quad F(\boldsymbol{\mu}) \qquad \longrightarrow \qquad \underset{\boldsymbol{\mu} \in \mathbb{R}^{n_{\boldsymbol{\mu}}}}{\text{minimize}} \quad m(\boldsymbol{\mu})$$

[2] Must be *computable* and apply to general, nonlinear PDEs

Proposed approach: managed inexactness

Replace expensive PDE with inexpensive approximation model

- **Reduced-order models** used for *inexact PDE evaluations*
- **Partially converged solutions** used for *inexact PDE evaluations*

$$\underset{\boldsymbol{\mu}\in\mathbb{R}^{n_{\boldsymbol{\mu}}}}{\text{minimize}} \quad F(\boldsymbol{\mu}) \qquad \longrightarrow \qquad \underset{\boldsymbol{\mu}\in\mathbb{R}^{n_{\boldsymbol{\mu}}}}{\text{minimize}} \quad m(\boldsymbol{\mu})$$

Manage inexactness with trust region method

- Embedded in globally convergent **trust region** method
- **Error indicators**[2] to account for *all* sources of inexactness
- **Refinement** of approximation model using *greedy algorithms*

$$\underset{\boldsymbol{\mu}\in\mathbb{R}^{n_{\boldsymbol{\mu}}}}{\text{minimize}} \quad F(\boldsymbol{\mu}) \qquad \longrightarrow \qquad \begin{array}{ll} \underset{\boldsymbol{\mu}\in\mathbb{R}^{n_{\boldsymbol{\mu}}}}{\text{minimize}} & m_k(\boldsymbol{\mu}) \\ \text{subject to} & ||\boldsymbol{\mu} - \boldsymbol{\mu}_k|| \leq \Delta_k \end{array}$$

[2] Must be *computable* and apply to general, nonlinear PDEs

Proposed approach: managed inexactness

Replace expensive PDE with inexpensive approximation model

- **Reduced-order models** used for *inexact PDE evaluations*
- **Partially converged solutions** used for *inexact PDE evaluations*

$$\underset{\boldsymbol{\mu}\in\mathbb{R}^{n_{\boldsymbol{\mu}}}}{\text{minimize}} \quad F(\boldsymbol{\mu}) \qquad \longrightarrow \qquad \underset{\boldsymbol{\mu}\in\mathbb{R}^{n_{\boldsymbol{\mu}}}}{\text{minimize}} \quad m(\boldsymbol{\mu})$$

Manage inexactness with trust region method

- Embedded in globally convergent **trust region** method
- **Error indicators**[2] to account for *all* sources of inexactness
- **Refinement** of approximation model using *greedy algorithms*

$$\underset{\boldsymbol{\mu}\in\mathbb{R}^{n_{\boldsymbol{\mu}}}}{\text{minimize}} \quad F(\boldsymbol{\mu}) \qquad \longrightarrow \qquad \begin{array}{ll} \underset{\boldsymbol{\mu}\in\mathbb{R}^{n_{\boldsymbol{\mu}}}}{\text{minimize}} & m_k(\boldsymbol{\mu}) \\ \text{subject to} & ||\boldsymbol{\mu} - \boldsymbol{\mu}_k|| \leq \Delta_k \end{array}$$

[2] Must be *computable* and apply to general, nonlinear PDEs

Relationship between the objective function and model

- First-order consistency [Alexandrov et al., 1998]

$$m_k(\boldsymbol{\mu}_k) = F(\boldsymbol{\mu}_k) \qquad \nabla m_k(\boldsymbol{\mu}_k) = \nabla F(\boldsymbol{\mu}_k)$$

- The Carter condition [Carter, 1989, Carter, 1991]

$$||\nabla F(\boldsymbol{\mu}_k) - \nabla m_k(\boldsymbol{\mu}_k)|| \leq \eta \, ||\nabla m_k(\boldsymbol{\mu}_k)|| \qquad \eta \in (0,\, 1)$$

- Asymptotic gradient bound [Heinkenschloss and Vicente, 2002]

$$||\nabla F(\boldsymbol{\mu}_k) - \nabla m_k(\boldsymbol{\mu}_k)|| \leq \xi \min\{||\nabla m_k(\boldsymbol{\mu}_k)|| \,,\, \Delta_k\} \qquad \xi > 0$$

Asymptotic gradient bound permits the use of an ***error indicator****:* φ_k

$$||\nabla F(\boldsymbol{\mu}) - \nabla m_k(\boldsymbol{\mu})|| \leq \xi \varphi_k(\boldsymbol{\mu}) \qquad \xi > 0$$
$$\varphi_k(\boldsymbol{\mu}_k) \leq \kappa_\varphi \min\{||\nabla m_k(\boldsymbol{\mu}_k)|| \,,\, \Delta_k\}$$

Trust region method with inexact gradients [Kouri et al., 2013]

1: **Model update**: Choose model m_k such that error indicator φ_k satisfies

$$\varphi_k(\boldsymbol{\mu}_k) \leq \kappa_\varphi \min\{||\nabla m_k(\boldsymbol{\mu}_k)||\,,\, \Delta_k\}$$

2: **Step computation**: Approximately solve the trust region subproblem

$$\hat{\boldsymbol{\mu}}_k \quad = \quad \underset{\boldsymbol{\mu}\in\mathbb{R}^{n_{\boldsymbol{\mu}}}}{\arg\min}\; m_k(\boldsymbol{\mu}) \quad \text{subject to} \quad ||\boldsymbol{\mu} - \boldsymbol{\mu}_k|| \leq \Delta_k$$

3: **Step acceptance**: Compute actual-to-predicted reduction

$$\rho_k = \frac{F(\boldsymbol{\mu}_k) - F(\hat{\boldsymbol{\mu}}_k)}{m_k(\boldsymbol{\mu}_k) - m_k(\hat{\boldsymbol{\mu}}_k)}$$

if $\rho_k \geq \eta_1$ **then** $\boldsymbol{\mu}_{k+1} = \hat{\boldsymbol{\mu}}_k$ **else** $\boldsymbol{\mu}_{k+1} = \boldsymbol{\mu}_k$ **end if**

4: **Trust region update**:

if $\rho_k \leq \eta_1$ **then** $\Delta_{k+1} \in (0, \gamma\,||\hat{\boldsymbol{\mu}}_k - \boldsymbol{\mu}_k||]$ **end if**

if $\rho_k \in (\eta_1, \eta_2)$ **then** $\Delta_{k+1} \in [\gamma\,||\hat{\boldsymbol{\mu}}_k - \boldsymbol{\mu}_k||\,, \Delta_k]$ **end if**

if $\rho_k \geq \eta_2$ **then** $\Delta_{k+1} \in [\Delta_k, \Delta_{\max}]$ **end if**

Trust region method with inexact gradients [Kouri et al., 2013]

1: **Model update**: Choose model m_k such that error indicator φ_k satisfies

$$\varphi_k(\boldsymbol{\mu}_k) \leq \kappa_\varphi \min\{||\nabla m_k(\boldsymbol{\mu}_k)||\,,\ \Delta_k\}$$

2: **Step computation**: Approximately solve the trust region subproblem

$$\hat{\boldsymbol{\mu}}_k \quad = \quad \operatorname*{arg\,min}_{\boldsymbol{\mu}\in\mathbb{R}^{n_{\boldsymbol{\mu}}}} m_k(\boldsymbol{\mu}) \quad \text{subject to} \quad ||\boldsymbol{\mu} - \boldsymbol{\mu}_k|| \leq \Delta_k$$

3: **Step acceptance**: Compute actual-to-predicted reduction

$$\rho_k = \frac{F(\boldsymbol{\mu}_k) - F(\hat{\boldsymbol{\mu}}_k)}{m_k(\boldsymbol{\mu}_k) - m_k(\hat{\boldsymbol{\mu}}_k)}$$

if $\rho_k \geq \eta_1$ **then** $\boldsymbol{\mu}_{k+1} = \hat{\boldsymbol{\mu}}_k$ **else** $\boldsymbol{\mu}_{k+1} = \boldsymbol{\mu}_k$ **end if**

4: **Trust region update**:

if $\rho_k \leq \eta_1$ **then** $\Delta_{k+1} \in (0, \gamma\,||\hat{\boldsymbol{\mu}}_k - \boldsymbol{\mu}_k||]$ **end if**

if $\rho_k \in (\eta_1, \eta_2)$ **then** $\Delta_{k+1} \in [\gamma\,||\hat{\boldsymbol{\mu}}_k - \boldsymbol{\mu}_k||\,, \Delta_k]$ **end if**

if $\rho_k \geq \eta_2$ **then** $\Delta_{k+1} \in [\Delta_k, \Delta_{\max}]$ **end if**

Trust region ingredients for global convergence

Approximation model

$$m_k(\boldsymbol{\mu})$$

Error indicator

$$||\nabla F(\boldsymbol{\mu}) - \nabla m_k(\boldsymbol{\mu})|| \leq \xi \varphi_k(\boldsymbol{\mu}), \qquad \xi > 0$$

Adaptivity

$$\varphi_k(\boldsymbol{\mu}_k) \leq \kappa_\varphi \min\{||\nabla m_k(\boldsymbol{\mu}_k)||\,,\, \Delta_k\}$$

Global convergence

$$\liminf_{k\to\infty} \; ||\nabla F(\boldsymbol{\mu}_k)|| = 0$$

Source of inexactness/efficiency: projection-based model reduction

- Model reduction ansatz: *state vector lies in low-dimensional subspace*

$$\boldsymbol{u} \approx \boldsymbol{\Phi} \boldsymbol{u}_r$$

  - $\boldsymbol{\Phi} = \begin{bmatrix} \boldsymbol{\phi}^1 & \cdots & \boldsymbol{\phi}^{k_{\boldsymbol{u}}} \end{bmatrix} \in \mathbb{R}^{n_{\boldsymbol{u}} \times k_{\boldsymbol{u}}}$ is the reduced (trial) basis ($n_{\boldsymbol{u}} \gg k_{\boldsymbol{u}}$)
  - $\boldsymbol{u}_r \in \mathbb{R}^{k_{\boldsymbol{u}}}$ are the reduced coordinates of $\boldsymbol{u}$

- Substitute into $\boldsymbol{r}(\boldsymbol{u}, \boldsymbol{\mu}) = 0$ and project onto columnspace of a test basis $\boldsymbol{\Phi} \in \mathbb{R}^{n_{\boldsymbol{u}} \times k_{\boldsymbol{u}}}$ to obtain a square system

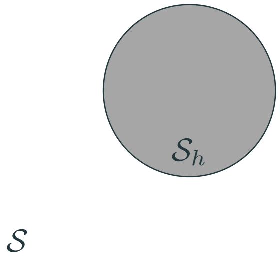
$$\boldsymbol{\Phi}^T \boldsymbol{r}(\boldsymbol{\Phi} \boldsymbol{u}_r, \boldsymbol{\mu}) = 0$$

Connection to finite element method: hierarchical subspaces

$\mathcal{S}$

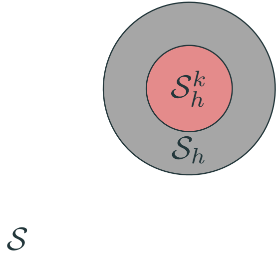
- $\mathcal{S}$ - infinite-dimensional trial space

Connection to finite element method: hierarchical subspaces



- $\mathcal{S}$ - infinite-dimensional trial space
- $\mathcal{S}_h$ - (large) finite-dimensional trial space

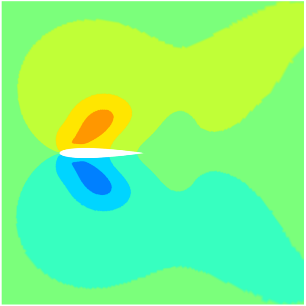
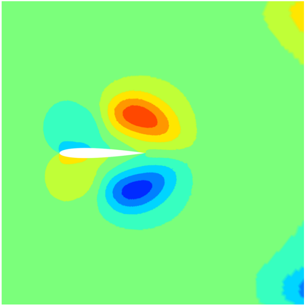
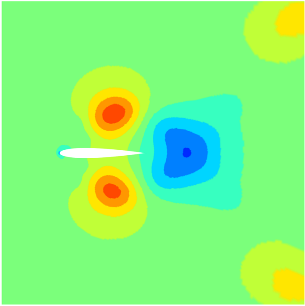
Connection to finite element method: hierarchical subspaces



- $\mathcal{S}$ - infinite-dimensional trial space
- $\mathcal{S}_h$ - (large) finite-dimensional trial space
- $\mathcal{S}_h^k$ - (small) finite-dimensional trial space
- $\mathcal{S}_h^k \subset \mathcal{S}_h \subset \mathcal{S}$

Few global, data-driven basis functions v. many local ones

- Instead of using traditional *local* shape functions, use **global shape functions**
- Instead of a-priori, analytical shape functions, leverage data-rich computing environment by using **data-driven modes**

Trust region method: ROM approximation model

Approximation models based on reduced-order models

$$m_k(\boldsymbol{\mu}) = \mathcal{J}(\boldsymbol{\Phi}_k \boldsymbol{u}_r(\boldsymbol{\mu}),\, \boldsymbol{\mu})$$

Error indicators from residual-based error bounds

$$\varphi_k(\boldsymbol{\mu}) = ||\boldsymbol{r}(\boldsymbol{\Phi}_k \boldsymbol{u}_r(\boldsymbol{\mu}),\, \boldsymbol{\mu})||_{\boldsymbol{\Theta}} + \left|\left|\boldsymbol{r}^{\boldsymbol{\lambda}}(\boldsymbol{\Phi}_k \boldsymbol{u}_r(\boldsymbol{\mu}),\, \boldsymbol{\Phi}_k \boldsymbol{\lambda}_r(\boldsymbol{\mu}),\, \boldsymbol{\mu})\right|\right|_{\boldsymbol{\Theta}^{\boldsymbol{\lambda}}}$$

Adaptivity to refine basis at trust region center

$$\boldsymbol{\Phi}_k = \begin{bmatrix} \boldsymbol{u}(\boldsymbol{\mu}_k) & \boldsymbol{\lambda}(\boldsymbol{\mu}_k) & \texttt{POD}(\boldsymbol{U}_k) & \texttt{POD}(\boldsymbol{V}_k) \end{bmatrix}$$

$$\boldsymbol{U}_k = \begin{bmatrix} \boldsymbol{u}(\boldsymbol{\mu}_0) & \cdots & \boldsymbol{u}(\boldsymbol{\mu}_{k-1}) \end{bmatrix} \qquad \boldsymbol{V}_k = \begin{bmatrix} \boldsymbol{\lambda}(\boldsymbol{\mu}_0) & \cdots & \boldsymbol{\lambda}(\boldsymbol{\mu}_{k-1}) \end{bmatrix}$$

Interpolation property of minimum-residual reduced-order models $\implies \varphi_k(\boldsymbol{\mu}_k) = 0$

Trust region method: ROM approximation model

Approximation models based on reduced-order models

$$m_k(\boldsymbol{\mu}) = \mathcal{J}(\boldsymbol{\Phi}_k \boldsymbol{u}_r(\boldsymbol{\mu}),\, \boldsymbol{\mu})$$

Error indicators from residual-based error bounds

$$\varphi_k(\boldsymbol{\mu}) = ||\boldsymbol{r}(\boldsymbol{\Phi}_k \boldsymbol{u}_r(\boldsymbol{\mu}),\, \boldsymbol{\mu})||_{\boldsymbol{\Theta}} + \left|\left|\boldsymbol{r}^{\boldsymbol{\lambda}}(\boldsymbol{\Phi}_k \boldsymbol{u}_r(\boldsymbol{\mu}),\, \boldsymbol{\Phi}_k \boldsymbol{\lambda}_r(\boldsymbol{\mu}),\, \boldsymbol{\mu})\right|\right|_{\boldsymbol{\Theta}^{\boldsymbol{\lambda}}}$$

Adaptivity to refine basis at trust region center

$$\boldsymbol{\Phi}_k = \begin{bmatrix} \boldsymbol{u}(\boldsymbol{\mu}_k) & \boldsymbol{\lambda}(\boldsymbol{\mu}_k) & \texttt{POD}(\boldsymbol{U}_k) & \texttt{POD}(\boldsymbol{V}_k) \end{bmatrix}$$

$$\boldsymbol{U}_k = \begin{bmatrix} \boldsymbol{u}(\boldsymbol{\mu}_0) & \cdots & \boldsymbol{u}(\boldsymbol{\mu}_{k-1}) \end{bmatrix} \qquad \boldsymbol{V}_k = \begin{bmatrix} \boldsymbol{\lambda}(\boldsymbol{\mu}_0) & \cdots & \boldsymbol{\lambda}(\boldsymbol{\mu}_{k-1}) \end{bmatrix}$$

Interpolation property of minimum-residual reduced-order models $\implies \varphi_k(\boldsymbol{\mu}_k) = 0$

$$\liminf_{k\to\infty} \; ||\nabla \mathcal{J}(\boldsymbol{u}(\boldsymbol{\mu}_k),\, \boldsymbol{\mu}_k)|| = 0$$

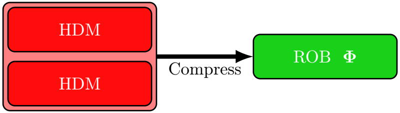
Trust region framework for optimization with ROMs

Schematic

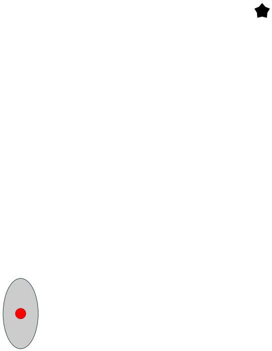
$\boldsymbol{\mu}$-space

Breakdown of Computational Effort

Trust region framework for optimization with ROMs



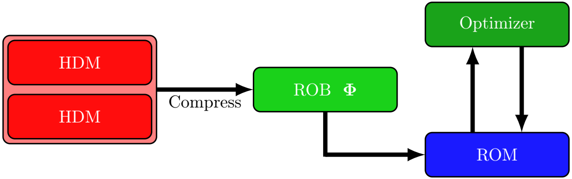
Schematic



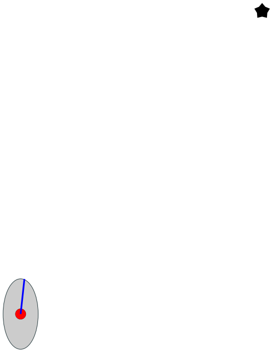
$\boldsymbol{\mu}$-space

Breakdown of Computational Effort

Trust region framework for optimization with ROMs



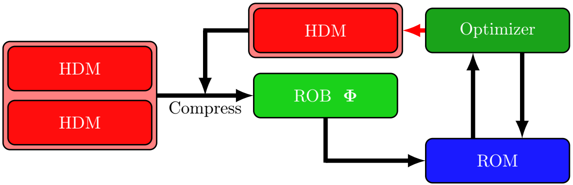
Schematic



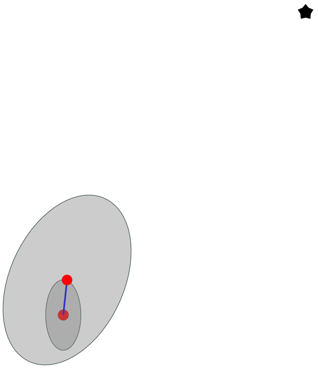
$\boldsymbol{\mu}$-space

Breakdown of Computational Effort

Trust region framework for optimization with ROMs



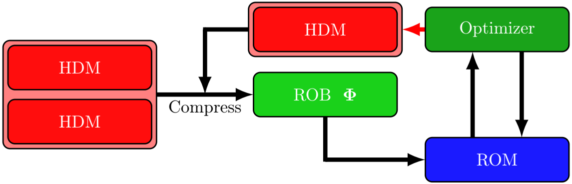
Schematic



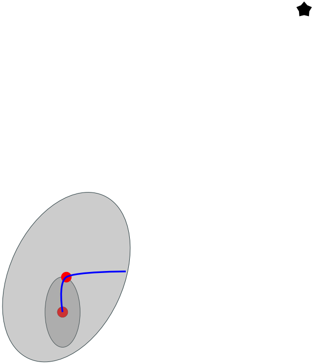
$\boldsymbol{\mu}$-space

Breakdown of Computational Effort

Trust region framework for optimization with ROMs



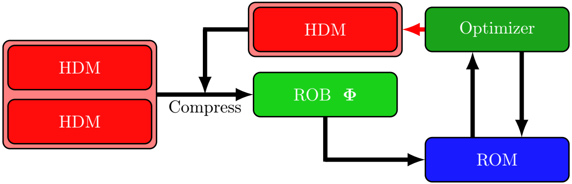
Schematic



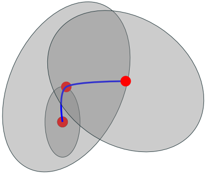
$\boldsymbol{\mu}$-space

Breakdown of Computational Effort

Trust region framework for optimization with ROMs



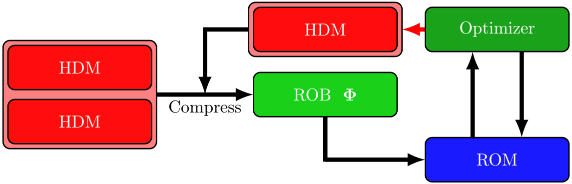
Schematic



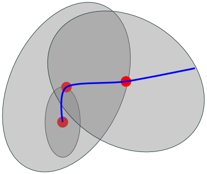
$\boldsymbol{\mu}$-space

Breakdown of Computational Effort

Trust region framework for optimization with ROMs



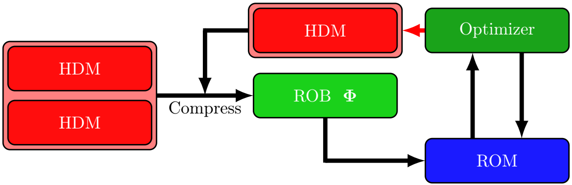
Schematic



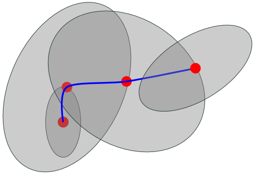
$\boldsymbol{\mu}$-space

Breakdown of Computational Effort

Trust region framework for optimization with ROMs



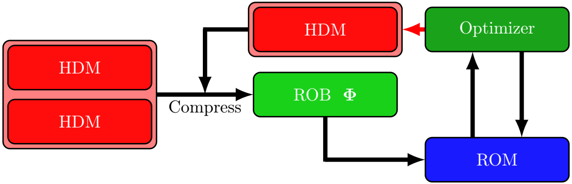
Schematic



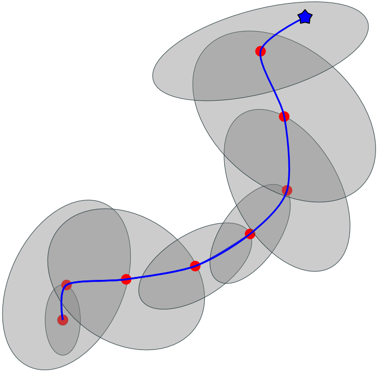
μ-space

Breakdown of Computational Effort

Trust region framework for optimization with ROMs



Schematic

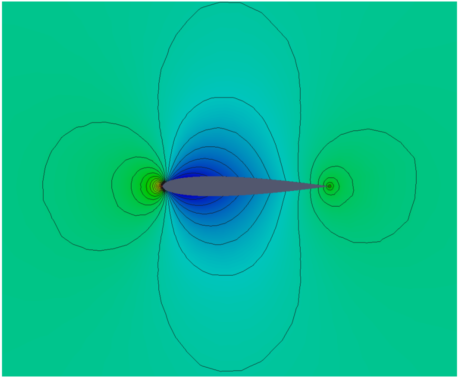


μ-space

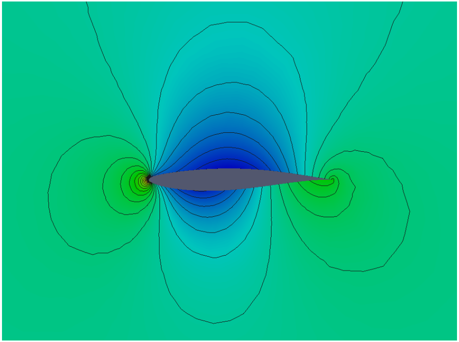
Breakdown of Computational Effort

Compressible, inviscid airfoil design

Pressure discrepancy minimization (Euler equations)



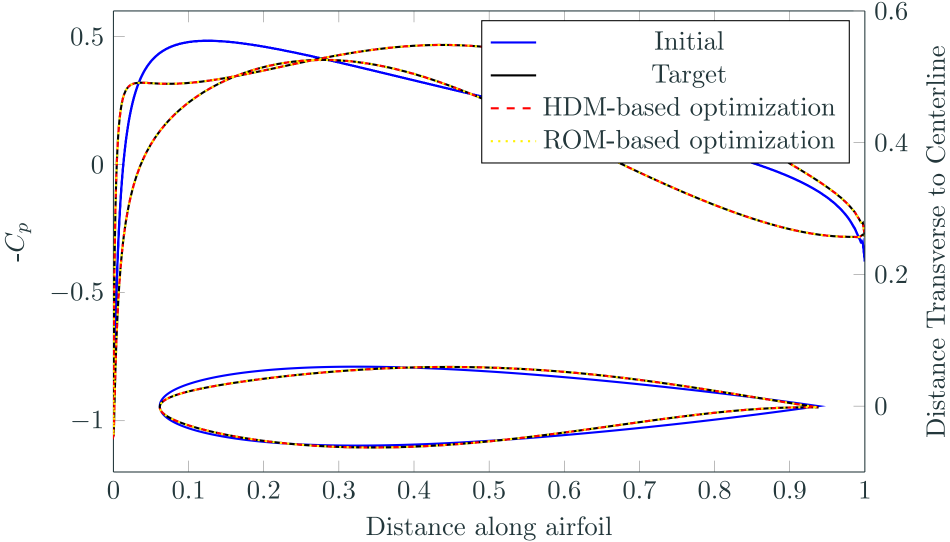
NACA0012: Initial



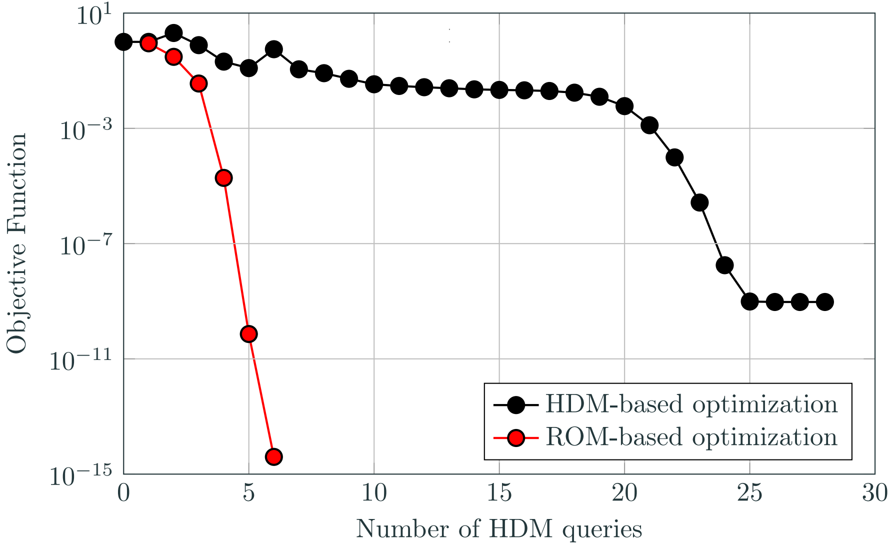
RAE2822: Target

Pressure field for airfoil configurations at $M_\infty = 0.5$, $\alpha = 0.0°$

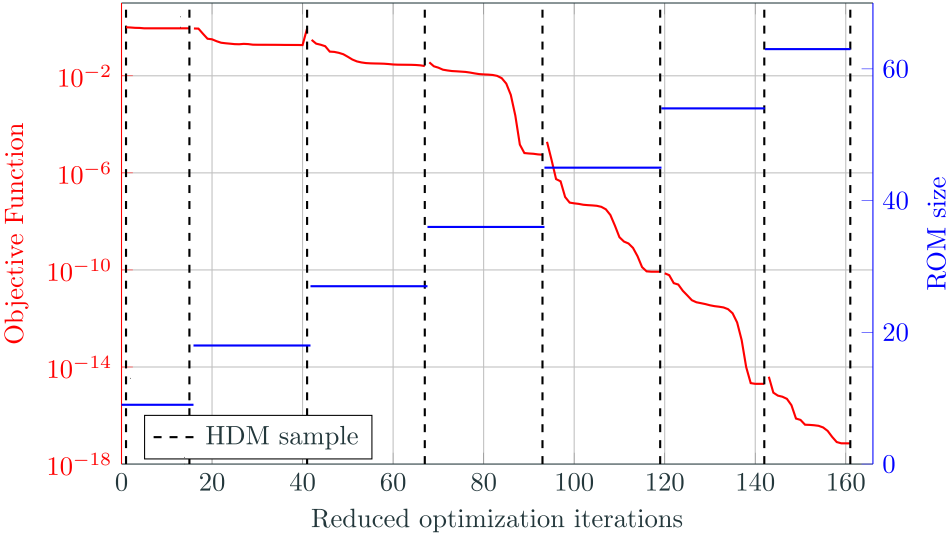
Proposed method: recovers target airfoil



Proposed method: 4× fewer HDM queries



At the cost of ROM queries

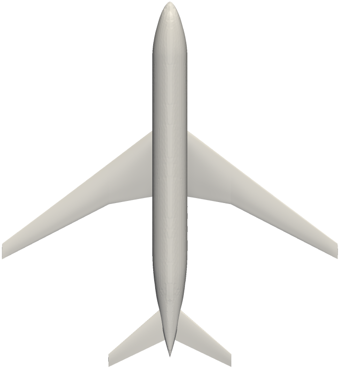


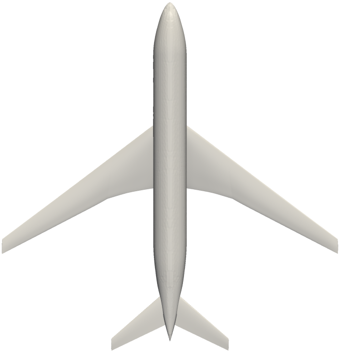
Shape optimization of aircraft in turbulent flow

$$\begin{aligned} \underset{\boldsymbol{\mu}\in\mathbb{R}^4}{\text{minimize}} \quad & -L_z(\boldsymbol{\mu})/L_x(\boldsymbol{\mu}) \\ \text{subject to} \quad & L_z(\boldsymbol{\mu}) = \bar{L}_z \end{aligned}$$

- **Flow**: $M = 0.85 \quad \alpha = 2.32^\circ \quad Re = 5 \times 10^6$
- **Equations**: RANS with Spalart-Allmaras
- **Solver**: Vertex-centered finite volume method
- **Mesh**: **11.5M** nodes, **68M** tetra, **69M** DOF

$$\boldsymbol{\mu} = [\mathbf{L} \quad r_x \quad \phi \quad r_z]$$



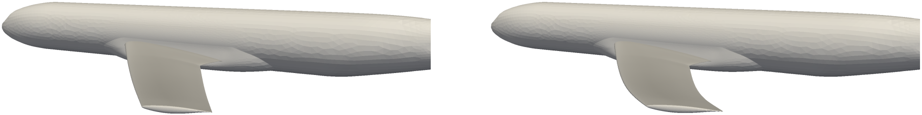


Shape optimization of aircraft in turbulent flow

$$\underset{\boldsymbol{\mu}\in\mathbb{R}^4}{\text{minimize}} \quad -L_z(\boldsymbol{\mu})/L_x(\boldsymbol{\mu})$$
$$\text{subject to} \quad L_z(\boldsymbol{\mu}) = \bar{L}_z$$

- **Flow**: $M = 0.85 \quad \alpha = 2.32^\circ \quad Re = 5 \times 10^6$
- **Equations**: RANS with Spalart-Allmaras
- **Solver**: Vertex-centered finite volume method
- **Mesh**: **11.5M** nodes, **68M** tetra, **69M** DOF

$$\boldsymbol{\mu} = [L \quad \mathbf{r_x} \quad \phi \quad r_z]$$



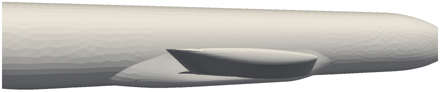
Localized sweep

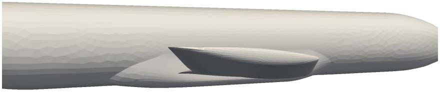
Shape optimization of aircraft in turbulent flow

$$\underset{\boldsymbol{\mu}\in\mathbb{R}^4}{\text{minimize}} \quad -L_z(\boldsymbol{\mu})/L_x(\boldsymbol{\mu})$$
$$\text{subject to} \quad L_z(\boldsymbol{\mu}) = \bar{L}_z$$

- **Flow**: $M = 0.85 \quad \alpha = 2.32^\circ \quad Re = 5 \times 10^6$
- **Equations**: RANS with Spalart-Allmaras
- **Solver**: Vertex-centered finite volume method
- **Mesh**: **11.5M** nodes, **68M** tetra, **69M** DOF

$$\boldsymbol{\mu} = [L \quad r_x \quad \boldsymbol{\phi} \quad r_z]$$





Twist

Shape optimization of aircraft in turbulent flow

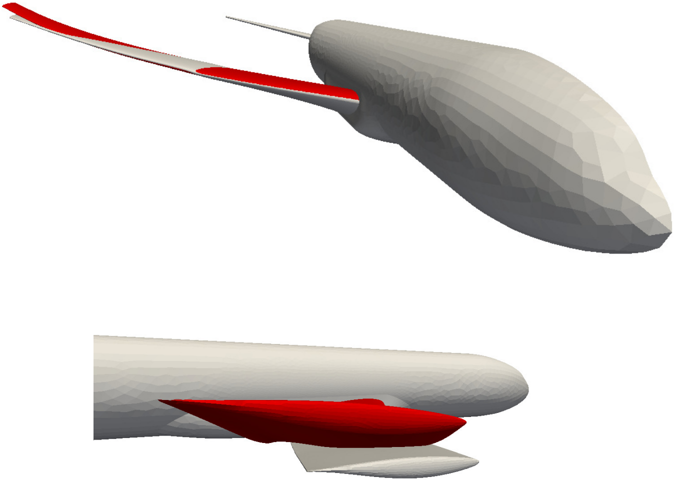
$$\underset{\boldsymbol{\mu}\in\mathbb{R}^4}{\text{minimize}} \quad -L_z(\boldsymbol{\mu})/L_x(\boldsymbol{\mu})$$
$$\text{subject to} \quad L_z(\boldsymbol{\mu}) = \bar{L}_z$$

- **Flow**: $M = 0.85 \quad \alpha = 2.32^\circ \quad Re = 5 \times 10^6$
- **Equations**: RANS with Spalart-Allmaras
- **Solver**: Vertex-centered finite volume method
- **Mesh**: **11.5M** nodes, **68M** tetra, **69M** DOF

$$\boldsymbol{\mu} = [L \quad r_x \quad \phi \quad \mathbf{r_z}]$$

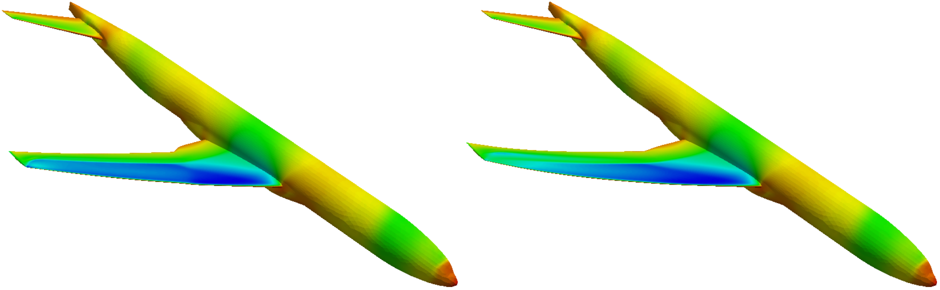
Localized dihedral

Optimized shape: reduction in 2.2 drag counts



Baseline (gray) and optimized shape (red) – 2× magnification

Optimized shape: reduction in 2.2 drag counts



Baseline (left) and optimized (right) shape – colored by C_p

Performance: ROM-TR method obtains same solution (to 4 digits of accuracy) as HDM-only optimization and only requires about 60% of the computation time.

Conclusion: Very promising results considering ROMs have notoriously poor prediction capabilities for problems with moving shocks/discontinuities.

Application to Bayesian parameter estimation

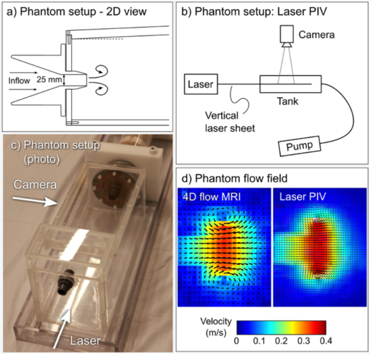
Enhance numerical simulation with noisy solution data

Let $\boldsymbol{z}$ denote noisy solution measurements that can be expressed as a function of the simulation parameters $\boldsymbol{\mu}$ and noise term $\boldsymbol{\epsilon}$ (known distribution) as

$$\boldsymbol{z} = \boldsymbol{h}(\boldsymbol{\mu}) + \boldsymbol{\epsilon},$$

where $\boldsymbol{h}$ is a function that maps simulation parameters to solution observations.

Example: Magnetic resonance imaging



Experimental setup

Noisy, low-resolution MRI data

Bayesian setting for parameter estimation

We want to estimate the probability distribution over the parameter space, given the data we have observed, i.e., the posterior $p(\boldsymbol{\mu}|\boldsymbol{z})$

$$p(\boldsymbol{\mu}|\boldsymbol{z}) \propto p(\boldsymbol{\mu})p(\boldsymbol{z}|\boldsymbol{\mu}),$$

where $p(\boldsymbol{\mu})$ is the prior distribution and the distribution $p(\boldsymbol{z}|\boldsymbol{\mu})$ can be inferred directly from our ansatz regarding the nature of the data ($\boldsymbol{z} = \boldsymbol{h}(\boldsymbol{\mu}) + \boldsymbol{\epsilon}$).

Importance sampling: empirical estimate of $p(\boldsymbol{\mu}|\boldsymbol{z})$ (and related statistics) where each sample assigned weights $(\boldsymbol{\mu}_j, w_j)$ to focus samples on important regions of parameter space, e.g., the expectation is approximated via the M-sample estimate

$$\mathbb{E}_M[\boldsymbol{g}(\boldsymbol{\mu})] = \sum_{j=1}^{M} \hat{w}_j \boldsymbol{g}(\boldsymbol{\mu}_j),$$

where $\hat{w}_j = \frac{w_j}{\sum_{j=1}^{M} w_j}$.

Parameter estimation via implicit sampling

Implicit sampling

Special case of importance sampling where samples computed by solving implicit equation [Morzfeld et al., 2015]

1) Find maximum a posteriori (MAP) point, $\boldsymbol{\mu}^*$, by maximizing

$$F(\boldsymbol{\mu}) = -\log p(\boldsymbol{\mu}) p(\boldsymbol{z}|\boldsymbol{\mu})$$

   $\rightarrow$ PDE-constrained optimization : $p(\boldsymbol{z}|\boldsymbol{\mu})$ requires solution of the PDE

2) Compute Hessian of F at $\boldsymbol{\mu}^*$, denoted $\boldsymbol{H}$
3) Implicit sampling in M random directions $\boldsymbol{\xi}_j$

$$F(\boldsymbol{\mu}^* + \lambda \boldsymbol{\xi}_j) - \phi = \frac{1}{2} \boldsymbol{\xi}_j^T \boldsymbol{H} \boldsymbol{\xi}_j$$

Acceleration using reduced-order models

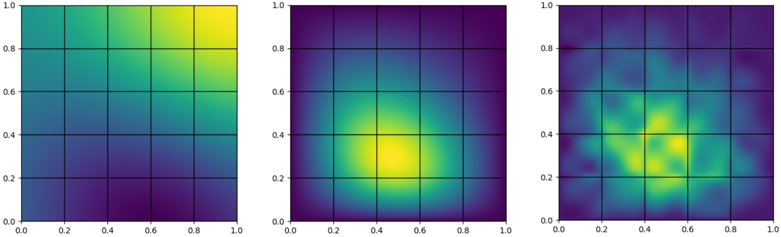
1) Accelerate optimization using trust-region framework and ROMs $\rightarrow \boldsymbol{\mu}^*, \boldsymbol{\Phi}$
2) Approximate Hessian using ROM and finite differences
3) Use ROM for implicit sampling

Parameter estimation: elliptic PDE

Consider the elliptic PDE, often used to model subsurface flow,

$$\begin{aligned} -\nabla \cdot (\kappa \nabla p) &= g \quad \text{in } \Omega \\ p &= h \quad \text{on } \partial\Omega, \end{aligned}$$

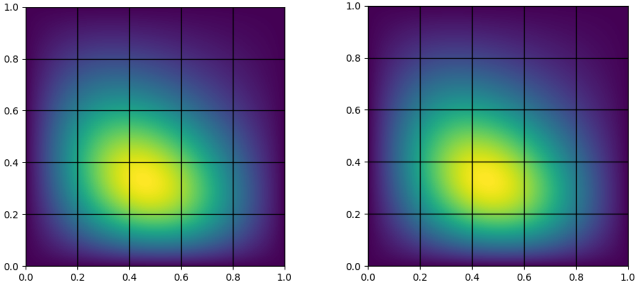
where p is the (partially observed) pressure field and κ is the (unknown) permeability. Pressure at 25% of FEM nodes is observed and the noise added is $\mathcal{N}(0, 0.3p_{\max})$.



True permeability (*left*), true pressure (*center*), and observed pressure (*right*).

Goal: *estimate the probability distribution of* κ *given the observations of* p

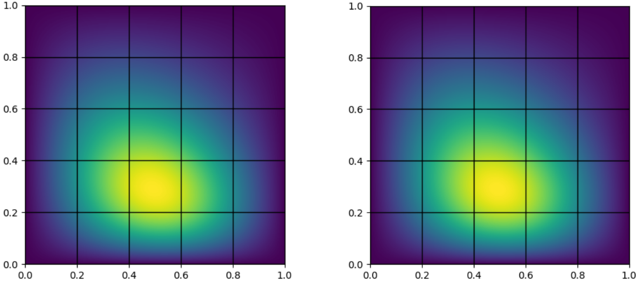
Computation of MAP point: HDM-only vs. HDM-ROM



MAP point: only HDM evaluations (*left*) and the ROM trust region method (*right*).

Performance:

	HDM-only	ROM-TR
HDM primal	27	8
HDM sensitivity	27	8
ROM primal	0	30
ROM sensitivity	0	30



Mean of posterior: only HDM evaluations (*left*) and the ROM trust region method (*right*).

Performance:

	Hessian evaluation		**Implicit sampling**	
	HDM-only	ROM-TR	HDM-only	ROM-TR
HDM primal	12	0	1799	0
HDM sensitivity	12	0	1799	0
ROM primal	0	12	0	1781
ROM sensitivity	0	12	0	1781

Leveraging inexactness to accelerate PDE-constrained optimization

- Framework introduced to accelerate PDE-constrained optimization
  - Adaptive *model reduction*
  - *Partially converged* primal and adjoint solutions
- Inexactness **managed** with flexible **trust region** method
- Applied to variety of problems in computational mechanics and outperforms standard methods
  - **5**× speedup: *subsonic* shape optimization of airfoil
  - **1.6**× speedup: *transonic* shape design of aircraft
- Extended/applied to accelerate **Bayesian parameter estimation**
  - Use ROM-TR method to find MAP point $\boldsymbol{\mu}^*$
  - Use reduced basis built during optimization to approximate Hessian at $\boldsymbol{\mu}^*$
  - Re-cast sampling procedure as optimization problem and apply ROM-TR

References I

Alexandrov, N. M., Dennis Jr, J. E., Lewis, R. M., and Torczon, V. (1998).
A trust-region framework for managing the use of approximation models in optimization.
Structural Optimization, 15(1):16–23.

Carter, R. G. (1989).
Numerical optimization in Hilbert space using inexact function and gradient evaluations.

Carter, R. G. (1991).
On the global convergence of trust region algorithms using inexact gradient information.
SIAM Journal on Numerical Analysis, 28(1):251–265.

Heinkenschloss, M. and Vicente, L. N. (2002).
Analysis of inexact trust-region SQP algorithms.
SIAM Journal on Optimization, 12(2):283–302.

References II

Kouri, D. P., Heinkenschloss, M., Ridzal, D., and van Bloemen Waanders, B. G. (2013).
A trust-region algorithm with adaptive stochastic collocation for PDE optimization under uncertainty.
SIAM Journal on Scientific Computing, 35(4):A1847–A1879.

Morzfeld, M., Tu, X., Wilkening, J., and Chorin, A. (2015).
Parameter estimation by implicit sampling.
Communications in Applied Mathematics and Computational Science, 10(2):205–225.