Optimization-based computational physics and high-order methods: from optimized analysis to design and data assimilation

Matthew J. Zahr[†] and Per-Olof Persson

CRD Postdoc Seminar Series

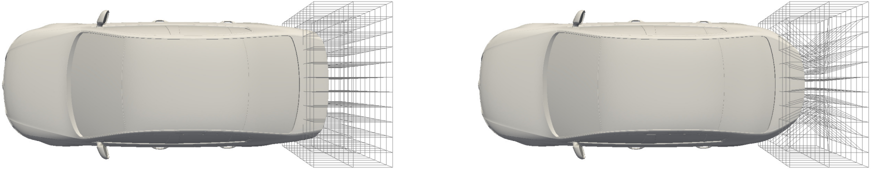
Lawrence Berkeley National Laboratory, Berkeley, CA

September 18, 2017

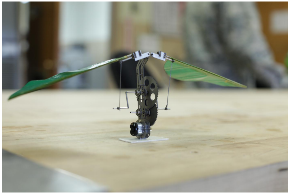
[†] Luis W. Alvarez Postdoctoral Fellow
Department of Mathematics
Lawrence Berkeley National Laboratory

PDE optimization is ubiquitous in science and engineering

Design: Find system that optimizes performance metric, satisfies constraints



Aerodynamic shape design of automobile

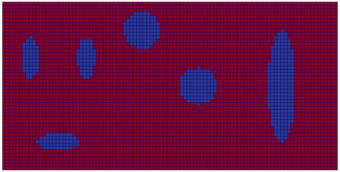


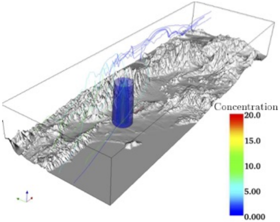
Optimal flapping motion of micro aerial vehicle

Control: Drive system to a desired state

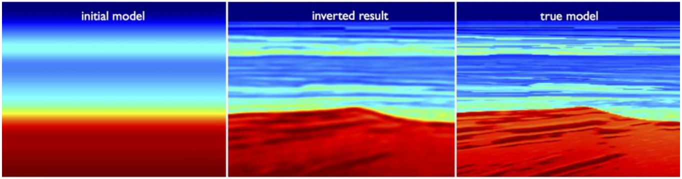
PDE optimization is ubiquitous in science and engineering

Inverse problems: Infer the problem setup given solution observations





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



Full waveform inversion – estimate subsurface of Earth's crust from acoustic measurements

Unsteady PDE-constrained optimization formulation

Goal: Find the solution of the *unsteady PDE-constrained optimization* problem

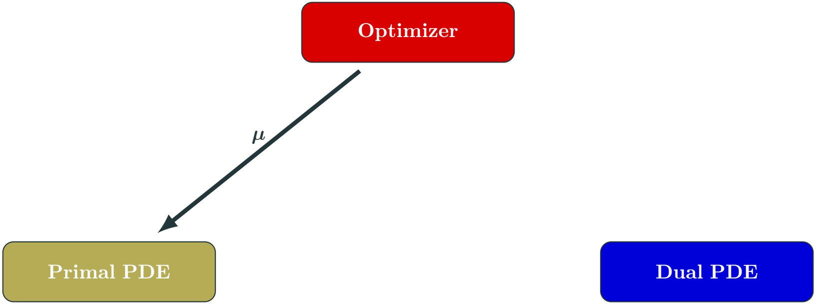
$$
\begin{aligned}
\underset{\boldsymbol{U},\ \boldsymbol{\mu}}{\text{minimize}} \quad & \mathcal{J}(\boldsymbol{U}, \boldsymbol{\mu}) \\
\text{subject to} \quad & \boldsymbol{C}(\boldsymbol{U}, \boldsymbol{\mu}) \leq 0 \\
& \frac{\partial \boldsymbol{U}}{\partial t} + \nabla \cdot \boldsymbol{F}(\boldsymbol{U}, \nabla \boldsymbol{U}) = 0 \quad \text{in} \quad v(\boldsymbol{\mu}, t)
\end{aligned}
$$

where

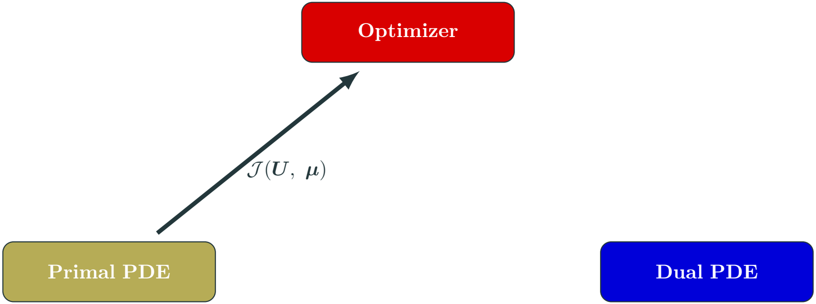
- $\boldsymbol{U}(\boldsymbol{x}, t)$ — PDE solution
- $\boldsymbol{\mu}$ — design/control parameters
- $\mathcal{J}(\boldsymbol{U}, \boldsymbol{\mu}) = \int_{T_0}^{T_f} \int_{\boldsymbol{\Gamma}} j(\boldsymbol{U}, \boldsymbol{\mu}, t)\, dS\, dt$ — objective function
- $\boldsymbol{C}(\boldsymbol{U}, \boldsymbol{\mu}) = \int_{T_0}^{T_f} \int_{\boldsymbol{\Gamma}} \mathbf{c}(\boldsymbol{U}, \boldsymbol{\mu}, t)\, dS\, dt$ — constraints

Nested approach to PDE-constrained optimization

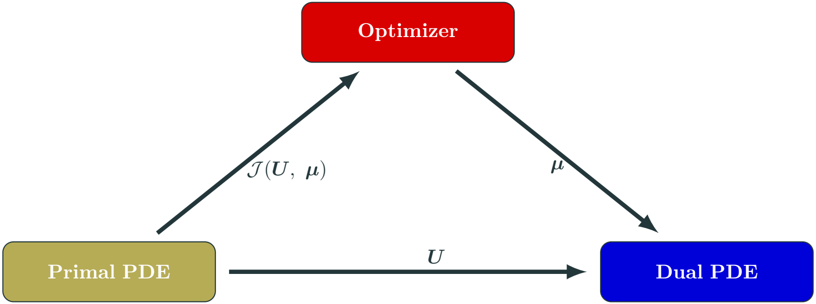
Nested approach to PDE-constrained optimization



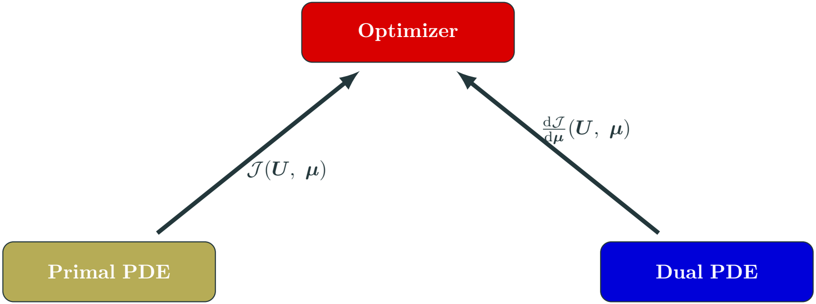
Nested approach to PDE-constrained optimization



Nested approach to PDE-constrained optimization



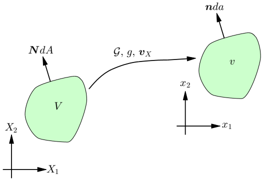
Nested approach to PDE-constrained optimization



Highlights of globally high-order discretization

- **Arbitrary Lagrangian-Eulerian** formulation:
  Map, $\mathcal{G}(\cdot, \boldsymbol{\mu}, t)$, from physical $v(\boldsymbol{\mu}, t)$ to reference V

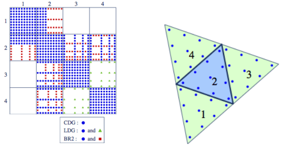
$$\left.\frac{\partial \boldsymbol{U_X}}{\partial t}\right|_{\boldsymbol{X}} + \nabla_{\boldsymbol{X}} \cdot \boldsymbol{F_X}(\boldsymbol{U_X},\ \nabla_{\boldsymbol{X}} \boldsymbol{U_X}) = 0$$



Mapping-Based ALE

- **Space discretization**: discontinuous Galerkin

$$\boldsymbol{M}\frac{\partial \boldsymbol{u}}{\partial t} = \boldsymbol{r}(\boldsymbol{u}, \boldsymbol{\mu}, t)$$



DG Discretization

- **Time discretization**: diagonally implicit RK

$$\boldsymbol{u}_n = \boldsymbol{u}_{n-1} + \sum_{i=1}^{s} b_i \boldsymbol{k}_{n,i}$$

$$\boldsymbol{M}\boldsymbol{k}_{n,i} = \Delta t_n \boldsymbol{r}\left(\boldsymbol{u}_{n,i},\ \boldsymbol{\mu},\ t_{n,i}\right)$$

$$\begin{array}{c|cccc} c_1 & a_{11} & & & \\ c_2 & a_{21} & a_{22} & & \\ \vdots & \vdots & \vdots & \ddots & \\ c_s & a_{s1} & a_{s2} & \cdots & a_{ss} \\ \hline & b_1 & b_2 & \cdots & b_s \end{array}$$

Butcher Tableau for DIRK

- **Quantity of interest**: solver-consistency

$$F(\boldsymbol{u}_0, \ldots, \boldsymbol{u}_{N_t}, \boldsymbol{k}_{1,1}, \ldots, \boldsymbol{k}_{N_t,s})$$

Adjoint method to efficiently compute gradients of QoI

- Consider the *fully discrete* output functional $F(\boldsymbol{u}_n, \boldsymbol{k}_{n,i}, \boldsymbol{\mu})$
    - Represents either the **objective** function or a **constraint**
- The *total derivative* with respect to the parameters $\boldsymbol{\mu}$, required in the context of gradient-based optimization, takes the form

$$\frac{\mathrm{d}F}{\mathrm{d}\boldsymbol{\mu}} = \frac{\partial F}{\partial \boldsymbol{\mu}} + \sum_{n=0}^{N_t} \frac{\partial F}{\partial \boldsymbol{u}_n} \frac{\partial \boldsymbol{u}_n}{\partial \boldsymbol{\mu}} + \sum_{n=1}^{N_t} \sum_{i=1}^{s} \frac{\partial F}{\partial \boldsymbol{k}_{n,i}} \frac{\partial \boldsymbol{k}_{n,i}}{\partial \boldsymbol{\mu}}$$

- The sensitivities, $\frac{\partial \boldsymbol{u}_n}{\partial \boldsymbol{\mu}}$ and $\frac{\partial \boldsymbol{k}_{n,i}}{\partial \boldsymbol{\mu}}$, are expensive to compute, requiring the solution of $n_{\boldsymbol{\mu}}$ linear evolution equations
- **Adjoint method**: alternative method for computing $\frac{\mathrm{d}F}{\mathrm{d}\boldsymbol{\mu}}$ that require one linear evolution equation for each quantity of interest, F

BERKELEY LAB

Dissection of fully discrete adjoint equations

- **Linear** evolution equations solved **backward** in time
- **Primal** state/stage, $\boldsymbol{u}_{n,i}$ required at each state/stage of dual problem
- Heavily dependent on **chosen ouput**

$$\boldsymbol{\lambda}_{N_t} = \frac{\partial F}{\partial \boldsymbol{u}_{N_t}}^T$$

$$\boldsymbol{\lambda}_{n-1} = \boldsymbol{\lambda}_n + \frac{\partial F}{\partial \boldsymbol{u}_{n-1}}^T + \sum_{i=1}^{s} \Delta t_n \frac{\partial \boldsymbol{r}}{\partial \boldsymbol{u}} \left(\boldsymbol{u}_{n,i},\ \boldsymbol{\mu},\ t_{n-1} + c_i \Delta t_n\right)^T \boldsymbol{\kappa}_{n,i}$$

$$\boldsymbol{M}^T \boldsymbol{\kappa}_{n,i} = \frac{\partial F}{\partial \boldsymbol{u}_{N_t}}^T + b_i \boldsymbol{\lambda}_n + \sum_{j=i}^{s} a_{ji} \Delta t_n \frac{\partial \boldsymbol{r}}{\partial \boldsymbol{u}} \left(\boldsymbol{u}_{n,j},\ \boldsymbol{\mu},\ t_{n-1} + c_j \Delta t_n\right)^T \boldsymbol{\kappa}_{n,j}$$

- Gradient reconstruction via dual variables

$$\frac{\mathrm{d}F}{\mathrm{d}\boldsymbol{\mu}} = \frac{\partial F}{\partial \boldsymbol{\mu}} + \boldsymbol{\lambda}_0{}^T \frac{\partial \boldsymbol{g}}{\partial \boldsymbol{\mu}}(\boldsymbol{\mu}) + \sum_{n=1}^{N_t} \Delta t_n \sum_{i=1}^{s} \boldsymbol{\kappa}_{n,i}{}^T \frac{\partial \boldsymbol{r}}{\partial \boldsymbol{\mu}}(\boldsymbol{u}_{n,i},\ \boldsymbol{\mu},\ t_{n,i})$$

[Zahr and Persson, 2016]

BERKELEY LAB

Optimal control, time-morphed geometry

Optimal Rigid Body Motion (RBM) and Time-Morphed Geometry (TMG), thrust $= 2.5$

Initial Guess	Optimal RBM	Optimal RBM/TMG
Energy $= 9.4096$	Energy $= 4.9476$	Energy $= 4.6182$
Thrust $= 0.1766$	Thrust $= 2.500$	Thrust $= 2.500$
	$T_x = 2.5$	$T_x = 2.5$

Energetically optimal flapping in three-dimensions

Energy = 1.4459e-01
Thrust = -1.1192e-01

Energy = 3.1378e-01
Thrust = 0.0000e+00

BERKELEY LAB

Extension: Parametrized time domain [Wang et al., 2017]

- Parametrization of time domain, e.g., flapping frequency, leads to parametrization of time discretization in fully discrete setting

$$T(\boldsymbol{\mu}) = N_t \Delta t \implies N_t = N_t(\boldsymbol{\mu}) \text{ or } \Delta t = \Delta t(\boldsymbol{\mu})$$

- Choose $\Delta t = \Delta t(\boldsymbol{\mu})$ to avoid discrete changes
- Does not change adjoint equations themselves, only reconstruction of gradient from adjoint solution

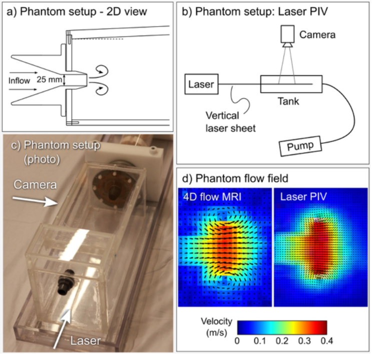
Energetically optimal flapping vs. required thrust

Initial Guess	Optimal $T_x = 0$	Optimal $T_x = 2.5$
$\text{Energy} = 1.8445$	$\text{Energy} = 0.21934$	$\text{Energy} = 6.2869$
$\text{Thrust} = 0.06729$	$\text{Thrust} = 0.0000$	$\text{Thrust} = 2.5000$

BERKELEY LAB

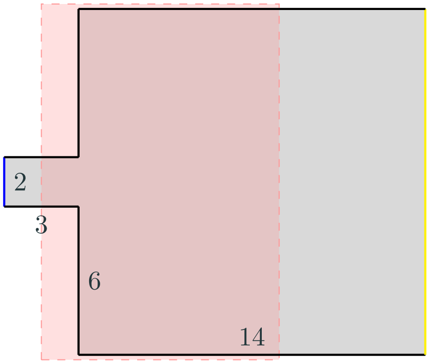
Super-resolution MR images through optimization

Space-time MRI data: noisy, low-resolution



- In collaboration with research team at Lund University, working to use our high-order optimization framework to generate "super-resolved" MR images
- *Idea*: Match CFD parameters (material properties, boundary conditions) to MRI data using optimization
- **Goal**: visualize high-resolution flow and accurately compute quantities of interest, i.e., wall shear stress

Phase I: synthetic data



Geometry and boundary conditions for synthetic MRI data assimilation setting. Boundary conditions: viscous wall (——), parametrized inflow (——), and outflow (——). MRI data collected in the red shaded region.

Coarse MRI grid (24×36), 10 time samples, 3% noise

Reconstructed flow

Synthetic MRI data $\boldsymbol{d}^*_{i,n}$ (top) and computational representation of MRI data $\boldsymbol{d}_{i,n}$ (bottom)

Fine MRI grid (40×60), 20 time samples, 10% noise

Reconstructed flow

Synthetic MRI data $\boldsymbol{d}^*_{i,n}$ (top) and computational representation of MRI data $\boldsymbol{d}_{i,n}$ (bottom)

Stochastic PDE-constrained optimization formulation

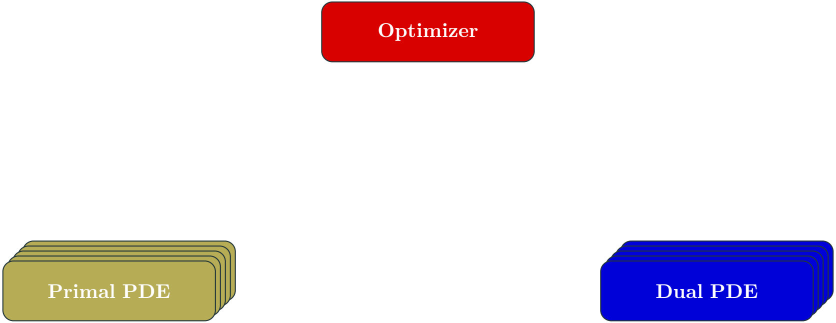
$$\begin{aligned}\underset{\boldsymbol{\mu}\in\mathbb{R}^{n_{\boldsymbol{\mu}}}}{\text{minimize}} \quad & \mathbb{E}[\mathcal{J}(\boldsymbol{u},\, \boldsymbol{\mu},\, \cdot\,)] \\ \text{subject to} \quad & \boldsymbol{r}(\boldsymbol{u};\, \boldsymbol{\mu},\, \boldsymbol{\xi}) = 0 \quad \forall \boldsymbol{\xi} \in \boldsymbol{\Xi}\end{aligned}$$

- $\boldsymbol{r} : \mathbb{R}^{n_{\boldsymbol{u}}} \times \mathbb{R}^{n_{\boldsymbol{\mu}}} \times \mathbb{R}^{n_{\boldsymbol{\xi}}} \rightarrow \mathbb{R}^{n_{\boldsymbol{u}}}$ — discretized stochastic PDE
- $\mathcal{J} : \mathbb{R}^{n_{\boldsymbol{u}}} \times \mathbb{R}^{n_{\boldsymbol{\mu}}} \times \mathbb{R}^{n_{\boldsymbol{\xi}}} \rightarrow \mathbb{R}$ — quantity of interest
- $\boldsymbol{u} \in \mathbb{R}^{n_{\boldsymbol{u}}}$ — PDE state vector
- $\boldsymbol{\mu} \in \mathbb{R}^{n_{\boldsymbol{\mu}}}$ — (deterministic) optimization parameters
- $\boldsymbol{\xi} \in \mathbb{R}^{n_{\boldsymbol{\xi}}}$ — stochastic parameters
- $\mathbb{E}[\mathcal{F}] \equiv \int_{\boldsymbol{\Xi}} \mathcal{F}(\boldsymbol{\xi})\rho(\boldsymbol{\xi})\, d\boldsymbol{\xi}$

BERKELEY LAB

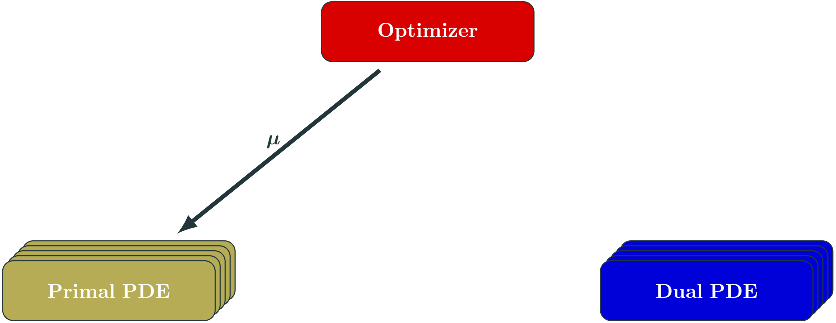
Nested approach to stochastic PDE-constrained optimization

Ensemble *of primal/dual PDE solves required at* ***every*** *optimization iteration*



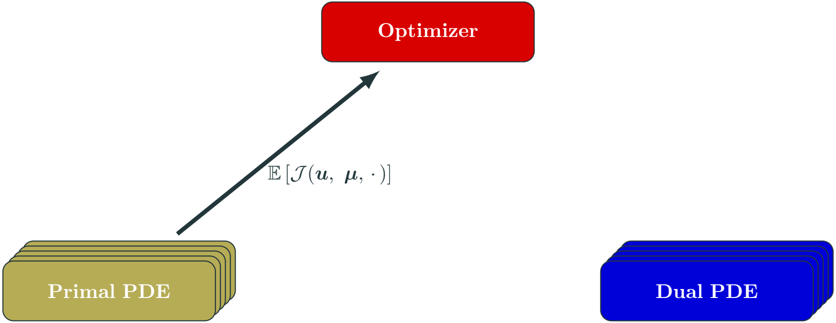
Nested approach to stochastic PDE-constrained optimization

Ensemble *of primal/dual PDE solves required at* ***every*** *optimization iteration*



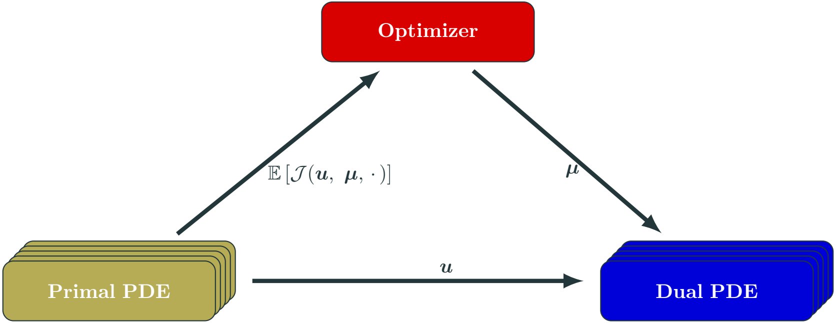
Nested approach to stochastic PDE-constrained optimization

Ensemble *of primal/dual PDE solves required at* ***every*** *optimization iteration*



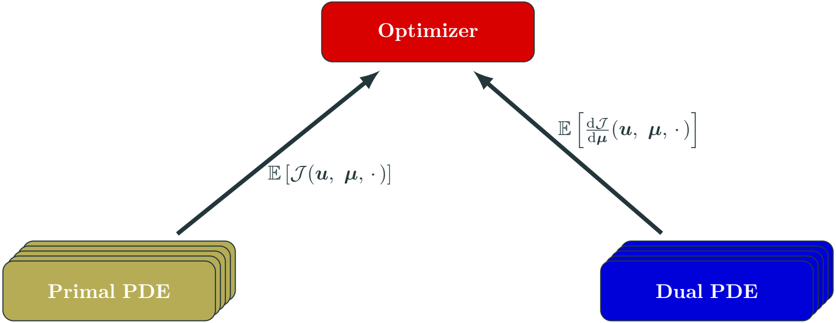
Nested approach to stochastic PDE-constrained optimization

Ensemble *of primal/dual PDE solves required at* ***every*** *optimization iteration*



Nested approach to stochastic PDE-constrained optimization

Ensemble *of primal/dual PDE solves required at* ***every*** *optimization iteration*



Proposed approach: managed inexactness

Replace expensive PDE with inexpensive approximation model

- **Anisotropic sparse grids** used for *inexact integration* of risk measures
- **Reduced-order models** used for *inexact PDE evaluations*

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

Proposed approach: managed inexactness

Replace expensive PDE with inexpensive approximation model

- **Anisotropic sparse grids** used for *inexact integration* of risk measures
- **Reduced-order models** used for *inexact PDE evaluations*

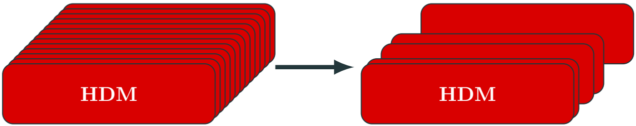
$$\underset{\boldsymbol{\mu}\in\mathbb{R}^{n_{\boldsymbol{\mu}}}}{\text{minimize}} \quad F(\boldsymbol{\mu}) \qquad \longrightarrow \qquad \underset{\boldsymbol{\mu}\in\mathbb{R}^{n_{\boldsymbol{\mu}}}}{\text{minimize}} \quad m(\boldsymbol{\mu})$$

Proposed approach: managed inexactness

Replace expensive PDE with inexpensive approximation model

- **Anisotropic sparse grids** used for *inexact integration* of risk measures
- **Reduced-order models** used for *inexact PDE evaluations*

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



Proposed approach: managed inexactness

Replace expensive PDE with inexpensive approximation model

- **Anisotropic sparse grids** used for *inexact integration* of risk measures
- **Reduced-order models** used for *inexact PDE evaluations*

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

Proposed approach: managed inexactness

Replace expensive PDE with inexpensive approximation model

- **Anisotropic sparse grids** used for *inexact integration* of risk measures
- **Reduced-order models** 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**[1] to account for *all* sources of inexactness
- **Refinement** of approximation model using *greedy algorithms*

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

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

First source of inexactness: anisotropic sparse grids

Stochastic collocation using anisotropic sparse grid nodes to approximate integral with summation

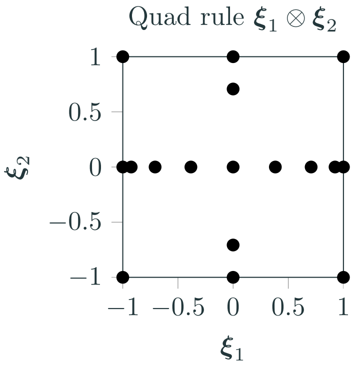
$$\begin{array}{ll} \underset{\boldsymbol{u}\in\mathbb{R}^{n_{\boldsymbol{u}}},\ \boldsymbol{\mu}\in\mathbb{R}^{n_{\boldsymbol{\mu}}}}{\text{minimize}} & \mathbb{E}[\mathcal{J}(\boldsymbol{u},\, \boldsymbol{\mu},\, \cdot\,)] \\ \text{subject to} & \boldsymbol{r}(\boldsymbol{u},\, \boldsymbol{\mu},\, \boldsymbol{\xi}) = 0 \quad \forall \boldsymbol{\xi} \in \boldsymbol{\Xi} \end{array}$$

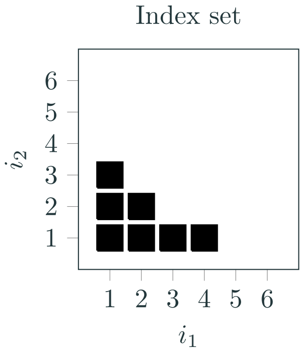
$$\Downarrow$$

$$\begin{array}{ll} \underset{\boldsymbol{u}\in\mathbb{R}^{n_{\boldsymbol{u}}},\ \boldsymbol{\mu}\in\mathbb{R}^{n_{\boldsymbol{\mu}}}}{\text{minimize}} & \mathbb{E}_{\mathcal{I}}[\mathcal{J}(\boldsymbol{u},\, \boldsymbol{\mu},\, \cdot\,)] \\ \text{subject to} & \boldsymbol{r}(\boldsymbol{u},\, \boldsymbol{\mu},\, \boldsymbol{\xi}) = 0 \quad \forall \boldsymbol{\xi} \in \boldsymbol{\Xi}_{\mathcal{I}} \end{array}$$

[Kouri et al., 2013, Kouri et al., 2014]

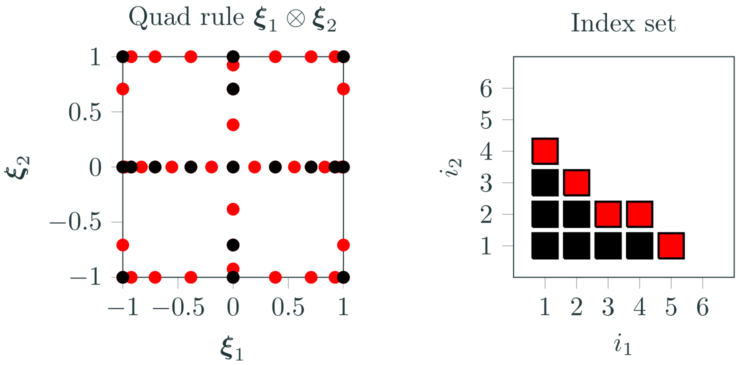
Source of inexactness: anisotropic sparse grids





Index set ($\mathcal{I}$) – ● Neighbors ($\mathcal{N}(\mathcal{I})$) – ●

Source of inexactness: anisotropic sparse grids



Index set $(\mathcal{I})$ – ●   Neighbors $(\mathcal{N}(\mathcal{I}))$ – ●

Second source of inexactness: reduced-order models

Stochastic collocation of the reduced-order model over anisotropic sparse grid nodes used to approximate integral with cheap summation

$$\begin{array}{ll} \underset{\boldsymbol{u}\in\mathbb{R}^{n_{\boldsymbol{u}}},\ \boldsymbol{\mu}\in\mathbb{R}^{n_{\boldsymbol{\mu}}}}{\text{minimize}} & \mathbb{E}[\mathcal{J}(\boldsymbol{u},\,\boldsymbol{\mu},\,\cdot\,)] \\ \text{subject to} & \boldsymbol{r}(\boldsymbol{u},\,\boldsymbol{\mu},\,\boldsymbol{\xi}) = 0 \quad \forall \boldsymbol{\xi} \in \boldsymbol{\Xi} \end{array}$$

$$\Downarrow$$

$$\begin{array}{ll} \underset{\boldsymbol{u}\in\mathbb{R}^{n_{\boldsymbol{u}}},\ \boldsymbol{\mu}\in\mathbb{R}^{n_{\boldsymbol{\mu}}}}{\text{minimize}} & \mathbb{E}_{\mathcal{I}}[\mathcal{J}(\boldsymbol{u},\,\boldsymbol{\mu},\,\cdot\,)] \\ \text{subject to} & \boldsymbol{r}(\boldsymbol{u},\,\boldsymbol{\mu},\,\boldsymbol{\xi}) = 0 \quad \forall \boldsymbol{\xi} \in \boldsymbol{\Xi}_{\mathcal{I}} \end{array}$$

$$\Downarrow$$

$$\begin{array}{ll} \underset{\boldsymbol{u}_r\in\mathbb{R}^{k_{\boldsymbol{u}}},\ \boldsymbol{\mu}\in\mathbb{R}^{n_{\boldsymbol{\mu}}}}{\text{minimize}} & \mathbb{E}_{\mathcal{I}}[\mathcal{J}(\boldsymbol{\Phi}\boldsymbol{u}_r,\,\boldsymbol{\mu},\,\cdot\,)] \\ \text{subject to} & \boldsymbol{\Phi}^T\boldsymbol{r}(\boldsymbol{\Phi}\boldsymbol{u}_r,\,\boldsymbol{\mu},\,\boldsymbol{\xi}) = 0 \quad \forall \boldsymbol{\xi} \in \boldsymbol{\Xi}_{\mathcal{I}} \end{array}$$

Source of inexactness: projection-based model reduction

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

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

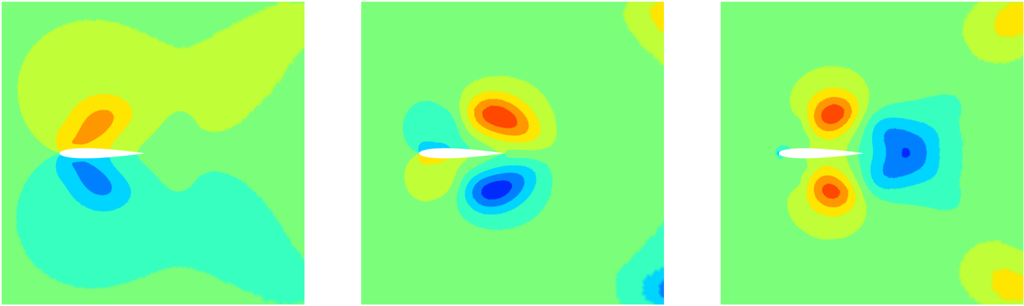
  - $\boldsymbol{\Phi} = \begin{bmatrix} \boldsymbol{\phi}^1 & \cdots & \boldsymbol{\phi}^{k_{\boldsymbol{u}}} \end{bmatrix} \in \mathbb{R}^{n_{\boldsymbol{u}} \times k_{\boldsymbol{u}}}$ is the reduced (trial) basis ($n_{\boldsymbol{u}} \gg k_{\boldsymbol{u}}$)
  - $\boldsymbol{u}_r \in \mathbb{R}^{k_{\boldsymbol{u}}}$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

- Substitute into $\boldsymbol{r}(\boldsymbol{u}, \boldsymbol{\mu}) = 0$ and perform Galerkin projection

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

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

- Instead of using traditional *local* shape functions (e.g., FEM), use *global* shape functions
- Instead of a-priori, analytical shape functions, leverage data-rich computing environment by using *data-driven* modes



Proposed approach: managed inexactness

Replace expensive PDE with inexpensive approximation model

- **Anisotropic sparse grids** used for *inexact integration* of risk measures
- **Reduced-order models** 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

Trust region ingredients for global convergence

Approximation models

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

Error indicators

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

Adaptivity

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

Global convergence

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

BERKELEY LAB

Trust region method: ROM/SG approximation model

Approximation models built on two sources of inexactness

$$m_k(\boldsymbol{\mu}) = \quad \mathbb{E}_{\mathcal{I}_k}\left[\mathcal{J}(\boldsymbol{\Phi}_k \boldsymbol{u}_r(\boldsymbol{\mu},\,\cdot\,),\,\boldsymbol{\mu},\,\cdot\,)\right]$$

Error indicators that account for both sources of error

$$\varphi_k(\boldsymbol{\mu}) = \alpha_1 \mathcal{E}_1(\boldsymbol{\mu};\,\mathcal{I}_k,\,\boldsymbol{\Phi}_k) + \alpha_2 \mathcal{E}_2(\boldsymbol{\mu};\,\mathcal{I}_k,\,\boldsymbol{\Phi}_k) + \alpha_3 \mathcal{E}_4(\boldsymbol{\mu};\,\mathcal{I}_k,\,\boldsymbol{\Phi}_k)$$

Reduced-order model errors

$$\mathcal{E}_1(\boldsymbol{\mu};\,\mathcal{I},\,\boldsymbol{\Phi}) = \mathbb{E}_{\mathcal{I}\cup\mathcal{N}(\mathcal{I})}\left[||\boldsymbol{r}(\boldsymbol{\Phi}\boldsymbol{u}_r(\boldsymbol{\mu},\,\cdot),\,\boldsymbol{\mu},\,\cdot\,)||\right]$$
$$\mathcal{E}_2(\boldsymbol{\mu};\,\mathcal{I},\,\boldsymbol{\Phi}) = \mathbb{E}_{\mathcal{I}\cup\mathcal{N}(\mathcal{I})}\left[\left|\left|\boldsymbol{r}^{\boldsymbol{\lambda}}(\boldsymbol{\Phi}\boldsymbol{u}_r(\boldsymbol{\mu},\,\cdot),\,\boldsymbol{\Phi}\boldsymbol{\lambda}_r(\boldsymbol{\mu},\,\cdot\,),\,\boldsymbol{\mu},\,\cdot\,)\right|\right|\right]$$

Sparse grid truncation errors

$$\mathcal{E}_4(\boldsymbol{\mu};\,\mathcal{I},\,\boldsymbol{\Phi}) = \mathbb{E}_{\mathcal{N}(\mathcal{I})}\left[||\nabla\mathcal{J}(\boldsymbol{\Phi}\boldsymbol{u}_r(\boldsymbol{\mu},\,\cdot\,),\,\boldsymbol{\mu},\,\cdot\,)||\right]$$

Adaptivity: Dimension-adaptive greedy method

while $\mathcal{E}_4(\mathbf{\Phi}, \mathcal{I}, \boldsymbol{\mu}_k) > \frac{\kappa_\varphi}{3\alpha_3} \min\{||\nabla m_k(\boldsymbol{\mu}_k)||\,, \Delta_k\}$ **do**

Refine index set: Dimension-adaptive sparse grids

$$\mathcal{I}_k \leftarrow \mathcal{I}_k \cup \{\mathbf{j}^*\} \qquad \text{where} \qquad \mathbf{j}^* = \arg\max_{\mathbf{j} \in \mathcal{N}(\mathcal{I}_k)} \mathbb{E}_{\mathbf{j}}\left[||\nabla \mathcal{J}(\mathbf{\Phi}\boldsymbol{u}_r(\boldsymbol{\mu}, \cdot), \boldsymbol{\mu}, \cdot)||\right]$$

Adaptivity: Dimension-adaptive greedy method

while $\mathcal{E}_4(\boldsymbol{\Phi}, \mathcal{I}, \boldsymbol{\mu}_k) > \frac{\kappa_\varphi}{3\alpha_3} \min\{||\nabla m_k(\boldsymbol{\mu}_k)||\,, \Delta_k\}$ **do**

Refine index set: Dimension-adaptive sparse grids

$$\mathcal{I}_k \leftarrow \mathcal{I}_k \cup \{\mathbf{j}^*\} \qquad \text{where} \qquad \mathbf{j}^* = \underset{\mathbf{j} \in \mathcal{N}(\mathcal{I}_k)}{\arg\max}\ \mathbb{E}_{\mathbf{j}}\left[||\nabla \mathcal{J}(\boldsymbol{\Phi}\boldsymbol{u}_r(\boldsymbol{\mu}, \cdot), \boldsymbol{\mu}, \cdot)||\right]$$

Refine reduced-order basis: Greedy sampling

while $\mathcal{E}_1(\boldsymbol{\Phi}, \mathcal{I}, \boldsymbol{\mu}_k) > \frac{\kappa_\varphi}{3\alpha_1} \min\{||\nabla m_k(\boldsymbol{\mu}_k)||\,, \Delta_k\}$ **do**

$$\boldsymbol{\Phi}_k \leftarrow \begin{bmatrix} \boldsymbol{\Phi}_k & \boldsymbol{u}(\boldsymbol{\mu}_k, \boldsymbol{\xi}^*) & \boldsymbol{\lambda}(\boldsymbol{\mu}_k, \boldsymbol{\xi}^*) \end{bmatrix}$$

$$\boldsymbol{\xi}^* = \underset{\boldsymbol{\xi} \in \boldsymbol{\Xi}_{\mathbf{j}^*}}{\arg\max}\ \rho(\boldsymbol{\xi})\,||\boldsymbol{r}(\boldsymbol{\Phi}_k \boldsymbol{u}_r(\boldsymbol{\mu}_k, \boldsymbol{\xi}), \boldsymbol{\mu}_k, \boldsymbol{\xi})||$$

end while

BERKELEY LAB

Adaptivity: Dimension-adaptive greedy method

while $\mathcal{E}_4(\boldsymbol{\Phi}, \mathcal{I}, \boldsymbol{\mu}_k) > \frac{\kappa_\varphi}{3\alpha_3} \min\{||\nabla m_k(\boldsymbol{\mu}_k)||\,, \Delta_k\}$ **do**

Refine index set: Dimension-adaptive sparse grids

$$\mathcal{I}_k \leftarrow \mathcal{I}_k \cup \{\mathbf{j}^*\} \qquad \text{where} \qquad \mathbf{j}^* = \operatorname*{arg\,max}_{\mathbf{j} \in \mathcal{N}(\mathcal{I}_k)} \mathbb{E}_{\mathbf{j}}\left[||\nabla \mathcal{J}(\boldsymbol{\Phi}\boldsymbol{u}_r(\boldsymbol{\mu}, \cdot), \boldsymbol{\mu}, \cdot)||\right]$$

Refine reduced-order basis: Greedy sampling

while $\mathcal{E}_1(\boldsymbol{\Phi}, \mathcal{I}, \boldsymbol{\mu}_k) > \frac{\kappa_\varphi}{3\alpha_1} \min\{||\nabla m_k(\boldsymbol{\mu}_k)||\,, \Delta_k\}$ **do**

$$\boldsymbol{\Phi}_k \leftarrow \begin{bmatrix} \boldsymbol{\Phi}_k & \boldsymbol{u}(\boldsymbol{\mu}_k, \boldsymbol{\xi}^*) & \boldsymbol{\lambda}(\boldsymbol{\mu}_k, \boldsymbol{\xi}^*) \end{bmatrix}$$

$$\boldsymbol{\xi}^* = \operatorname*{arg\,max}_{\boldsymbol{\xi} \in \boldsymbol{\Xi}_{\mathbf{j}^*}} \rho(\boldsymbol{\xi})\, ||\boldsymbol{r}(\boldsymbol{\Phi}_k \boldsymbol{u}_r(\boldsymbol{\mu}_k, \boldsymbol{\xi}), \boldsymbol{\mu}_k, \boldsymbol{\xi})||$$

end while

while $\mathcal{E}_2(\boldsymbol{\Phi}, \mathcal{I}, \boldsymbol{\mu}_k) > \frac{\kappa_\varphi}{3\alpha_2} \min\{||\nabla m_k(\boldsymbol{\mu}_k)||\,, \Delta_k\}$ **do**

$$\boldsymbol{\Phi}_k \leftarrow \begin{bmatrix} \boldsymbol{\Phi}_k & \boldsymbol{u}(\boldsymbol{\mu}_k, \boldsymbol{\xi}^*) & \boldsymbol{\lambda}(\boldsymbol{\mu}_k, \boldsymbol{\xi}^*) \end{bmatrix}$$

$$\boldsymbol{\xi}^* = \operatorname*{arg\,max}_{\boldsymbol{\xi} \in \boldsymbol{\Xi}_{\mathbf{j}^*}} \rho(\boldsymbol{\xi})\, \left|\left|\boldsymbol{r}^{\boldsymbol{\lambda}}(\boldsymbol{\Phi}_k \boldsymbol{u}_r(\boldsymbol{\mu}_k, \boldsymbol{\xi}), \boldsymbol{\Phi}_k \boldsymbol{\lambda}_r(\boldsymbol{\mu}_k, \boldsymbol{\xi}), \boldsymbol{\mu}_k, \boldsymbol{\xi})\right|\right|$$

end while

Optimal control of steady Burgers' equation

- **Optimization problem:**

$$\underset{\boldsymbol{\mu}\in\mathbb{R}^{n_{\boldsymbol{\mu}}}}{\text{minimize}} \quad \int_{\boldsymbol{\Xi}} \rho(\boldsymbol{\xi}) \left[\int_0^1 \frac{1}{2}(u(\boldsymbol{\mu}, \boldsymbol{\xi}, x) - \bar{u}(x))^2 \, dx + \frac{\alpha}{2} \int_0^1 z(\boldsymbol{\mu}, x)^2 \, dx \right] d\boldsymbol{\xi}$$

where $u(\boldsymbol{\mu}, \boldsymbol{\xi}, x)$ solves

$$-\nu(\boldsymbol{\xi})\partial_{xx}u(\boldsymbol{\mu}, \boldsymbol{\xi}, x) + u(\boldsymbol{\mu}, \boldsymbol{\xi}, x)\partial_x u(\boldsymbol{\mu}, \boldsymbol{\xi}, x) = z(\boldsymbol{\mu}, x) \quad x \in (0, 1), \quad \boldsymbol{\xi} \in \boldsymbol{\Xi}$$

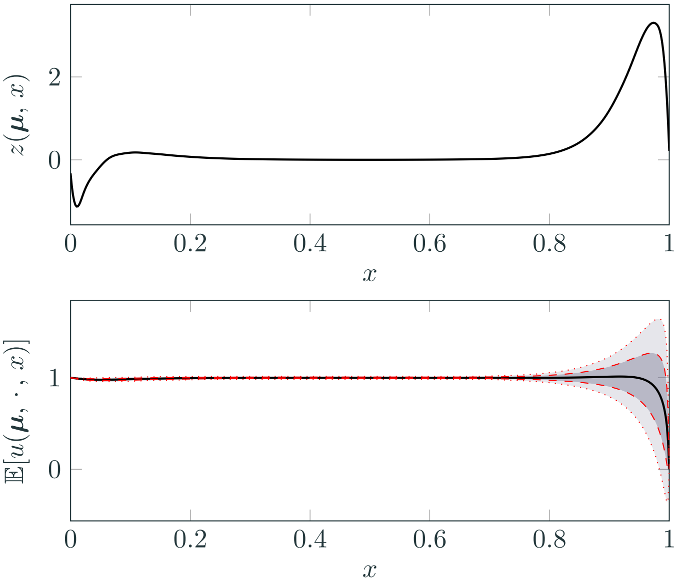
$$u(\boldsymbol{\mu}, \boldsymbol{\xi}, 0) = d_0(\boldsymbol{\xi}) \qquad u(\boldsymbol{\mu}, \boldsymbol{\xi}, 1) = d_1(\boldsymbol{\xi})$$

- **Target state:** $\bar{u}(x) \equiv 1$
- **Stochastic Space:** $\boldsymbol{\Xi} = [-1, 1]^3$, $\rho(\boldsymbol{\xi})d\boldsymbol{\xi} = 2^{-3}d\boldsymbol{\xi}$

$$\nu(\boldsymbol{\xi}) = 10^{\boldsymbol{\xi}_1 - 2} \qquad d_0(\boldsymbol{\xi}) = 1 + \frac{\boldsymbol{\xi}_2}{1000} \qquad d_1(\boldsymbol{\xi}) = \frac{\boldsymbol{\xi}_3}{1000}$$

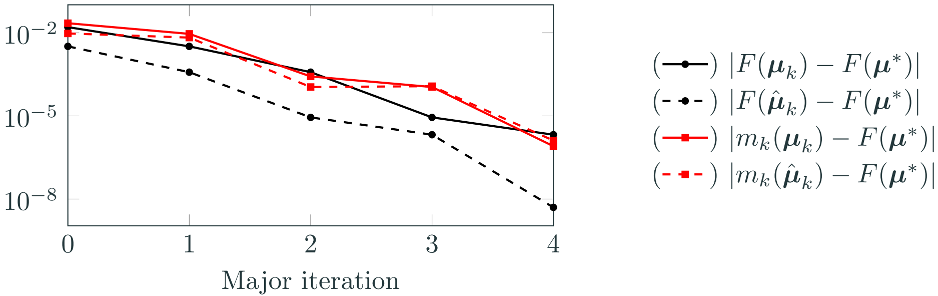
- **Parametrization:** $z(\boldsymbol{\mu}, x)$ – cubic splines with 51 knots, $n_{\boldsymbol{\mu}} = 53$

Optimal control and statistics



Optimal control and corresponding mean state (——) $\pm$ one (- - -) and two (⋯⋯) standard deviations

Global convergence without pointwise agreement

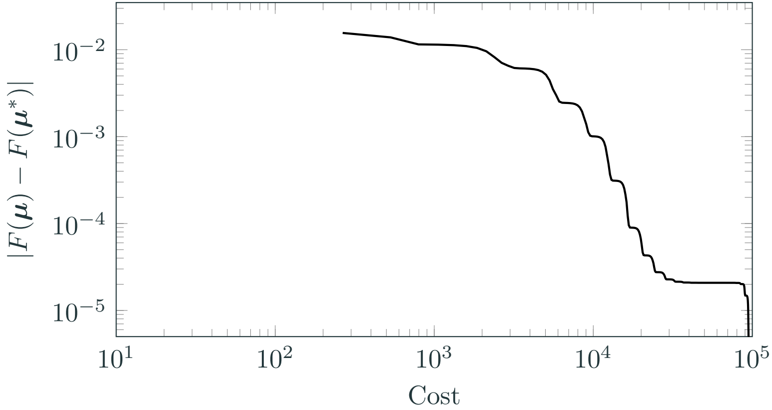


$F(\boldsymbol{\mu}_k)$	$m_k(\boldsymbol{\mu}_k)$	$F(\hat{\boldsymbol{\mu}}_k)$	$m_k(\hat{\boldsymbol{\mu}}_k)$	$\|\|\nabla F(\boldsymbol{\mu}_k)\|\|$	ρ_k	Success?
6.6506e-02	7.2694e-02	5.3655e-02	5.9922e-02	2.2959e-02	1.0257e+00	1.0000e+00
5.3655e-02	5.9593e-02	5.0783e-02	5.7152e-02	2.3424e-03	9.7512e-01	1.0000e+00
5.0783e-02	5.0670e-02	5.0412e-02	5.0292e-02	1.9724e-03	9.8351e-01	1.0000e+00
5.0412e-02	5.0292e-02	5.0405e-02	5.0284e-02	9.2654e-05	8.7479e-01	1.0000e+00
5.0405e-02	5.0404e-02	5.0403e-02	5.0401e-02	8.3139e-05	9.9946e-01	1.0000e+00
5.0403e-02	5.0401e-02	-	-	2.2846e-06	-	-

Convergence history of trust region method built on two-level approximation

Significant reduction in cost, if ROM only $10\times$ faster than HDM

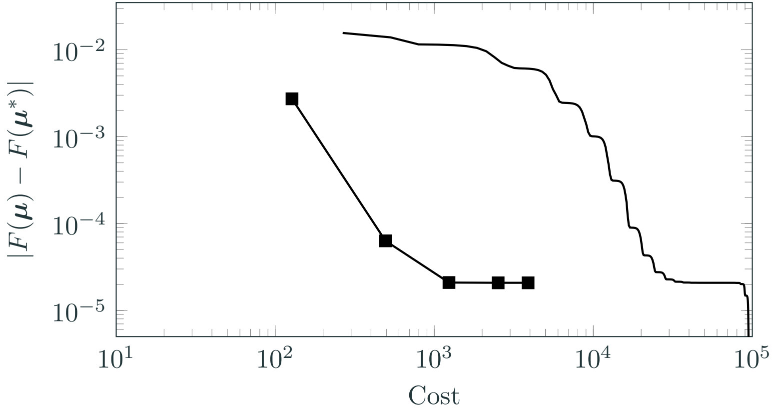
$$\text{Cost} = \text{nHdmPrim} + 0.5 \times \text{nHdmAdj} + \tau^{-1} \times (\text{nRomPrim} + 0.5 \times \text{nRomAdj})$$



5-level isotropic SG (——), dimension-adaptive SG [Kouri et al., 2014] ( ), and proposed ROM/SG for $\tau = 1$ ( ), $\tau = 10$ ( ), $\tau = 100$ ( ), $\tau = \infty$ ( )

Significant reduction in cost, if ROM only $10\times$ faster than HDM

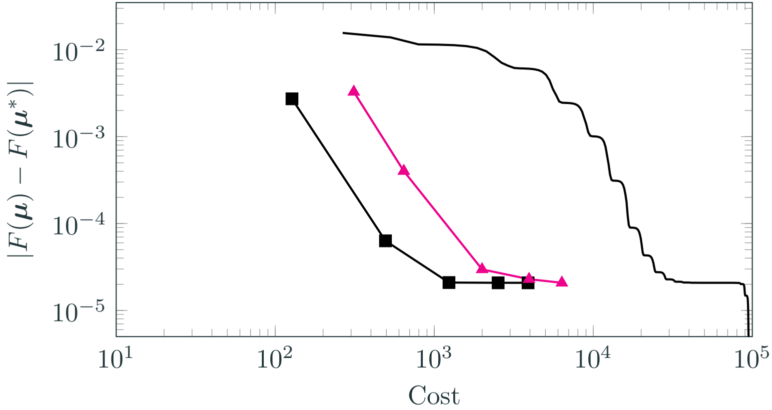
$$\text{Cost} = \text{nHdmPrim} + 0.5 \times \text{nHdmAdj} + \tau^{-1} \times (\text{nRomPrim} + 0.5 \times \text{nRomAdj})$$



5-level isotropic SG (——), dimension-adaptive SG [Kouri et al., 2014] (—■—), and proposed ROM/SG for $\tau = 1$ ( ), $\tau = 10$ ( ), $\tau = 100$ ( ), $\tau = \infty$ ( )

Significant reduction in cost, if ROM only $10\times$ faster than HDM

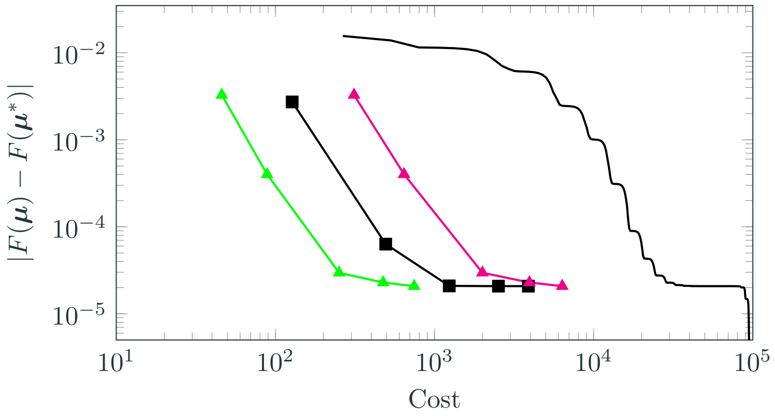
$$\text{Cost} = \text{nHdmPrim} + 0.5 \times \text{nHdmAdj} + \tau^{-1} \times (\text{nRomPrim} + 0.5 \times \text{nRomAdj})$$



5-level isotropic SG (—), dimension-adaptive SG [Kouri et al., 2014] (■), and proposed ROM/SG for $\tau = 1$ (▲), $\tau = 10$ ( ), $\tau = 100$ ( ), $\tau = \infty$ ( )

Significant reduction in cost, if ROM only $10\times$ faster than HDM

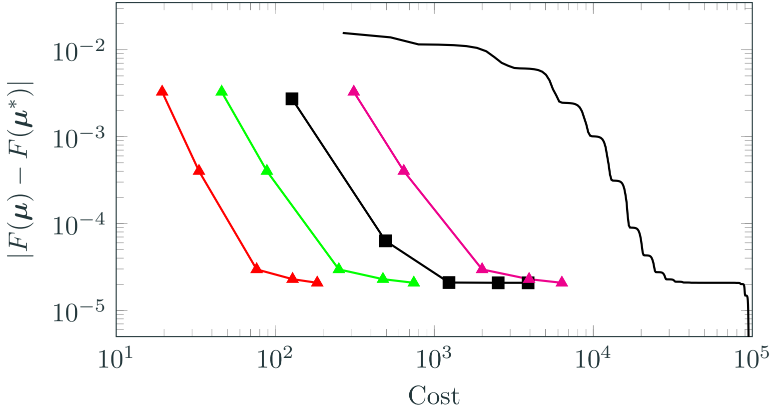
$$\text{Cost} = \text{nHdmPrim} + 0.5 \times \text{nHdmAdj} + \tau^{-1} \times (\text{nRomPrim} + 0.5 \times \text{nRomAdj})$$



5-level isotropic SG (——), dimension-adaptive SG [Kouri et al., 2014] (—■—), and proposed ROM/SG for $\tau = 1$ (—▲—), $\tau = 10$ (—▲—), $\tau = 100$ ( ), $\tau = \infty$ ( )

Significant reduction in cost, if ROM only $10\times$ faster than HDM

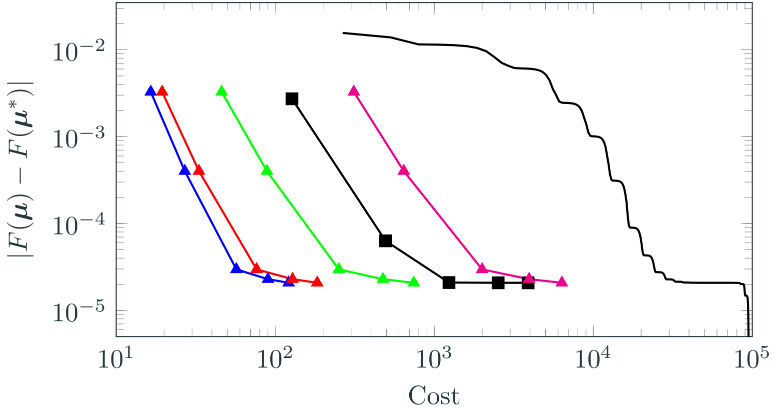
$$\text{Cost} = \text{nHdmPrim} + 0.5 \times \text{nHdmAdj} + \tau^{-1} \times (\text{nRomPrim} + 0.5 \times \text{nRomAdj})$$



5-level isotropic SG (—), dimension-adaptive SG [Kouri et al., 2014] (—■—), and proposed ROM/SG for $\tau = 1$ (—▲—), $\tau = 10$ (—▲—), $\tau = 100$ (—▲—), $\tau = \infty$ ( )

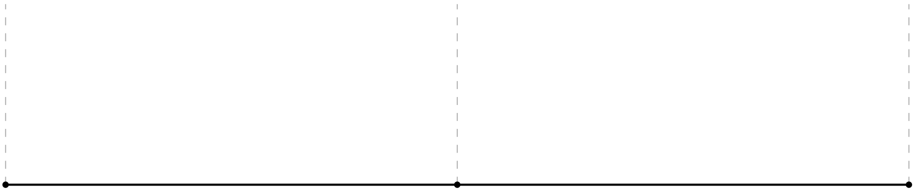
Significant reduction in cost, if ROM only $10\times$ faster than HDM

$$\text{Cost} = \text{nHdmPrim} + 0.5 \times \text{nHdmAdj} + \tau^{-1} \times (\text{nRomPrim} + 0.5 \times \text{nRomAdj})$$



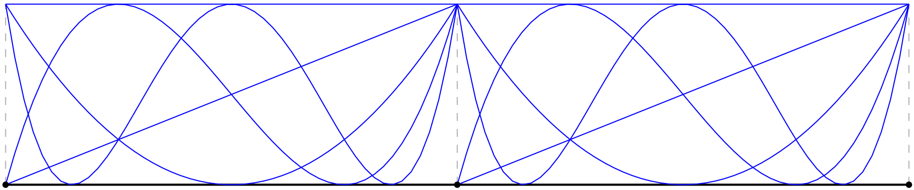
5-level isotropic SG (—), dimension-adaptive SG [Kouri et al., 2014] (—■—), and proposed ROM/SG for $\tau = 1$ (—▲—), $\tau = 10$ (—▲—), $\tau = 100$ (—▲—), $\tau = \infty$ (—▲—)

Optimization beyond design/control: high-order shock resolution



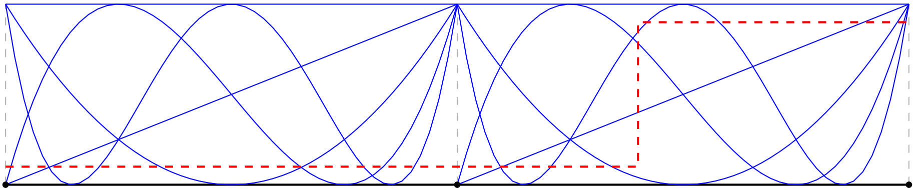
Fundamental issue: interpolate discontinuity with polynomial basis

Optimization beyond design/control: high-order shock resolution



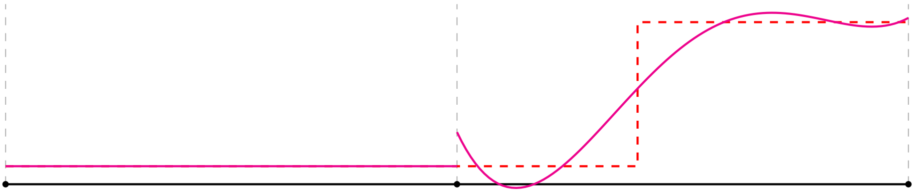
Fundamental issue: interpolate discontinuity with polynomial basis

Optimization beyond design/control: high-order shock resolution



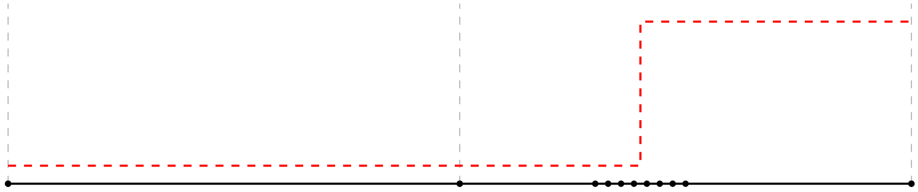
Fundamental issue: interpolate discontinuity with polynomial basis

Optimization beyond design/control: high-order shock resolution



Fundamental issue: interpolate discontinuity with polynomial basis

Optimization beyond design/control: high-order shock resolution

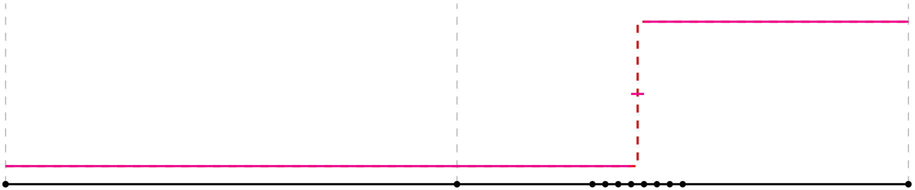


Fundamental issue: interpolate discontinuity with polynomial basis

Exising solutions: limiting, **adaptive refinement**, artificial viscosity

usually result in order reduction or very fine discretizations

Optimization beyond design/control: high-order shock resolution

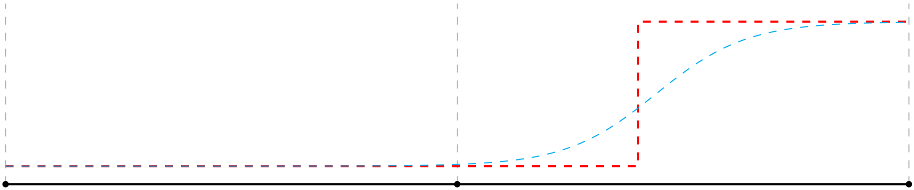


Fundamental issue: interpolate discontinuity with polynomial basis

Exising solutions: limiting, **adaptive refinement**, artificial viscosity

usually result in order reduction or very fine discretizations

Optimization beyond design/control: high-order shock resolution

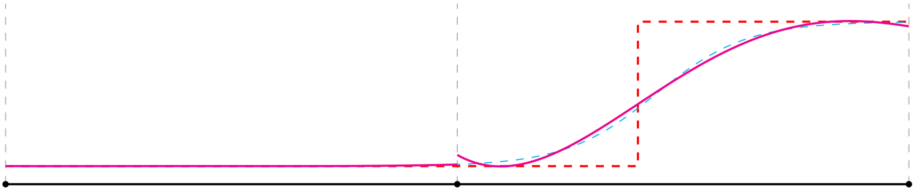


Fundamental issue: interpolate discontinuity with polynomial basis

Exising solutions: limiting, adaptive refinement, **artificial viscosity**

usually result in order reduction or very fine discretizations

Optimization beyond design/control: high-order shock resolution

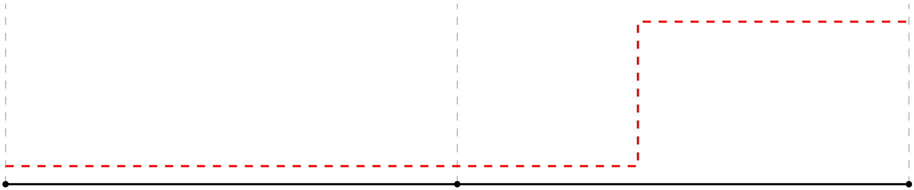


Fundamental issue: interpolate discontinuity with polynomial basis

Exising solutions: limiting, adaptive refinement, **artificial viscosity**

usually result in order reduction or very fine discretizations

Optimization beyond design/control: high-order shock resolution



Fundamental issue: interpolate discontinuity with polynomial basis

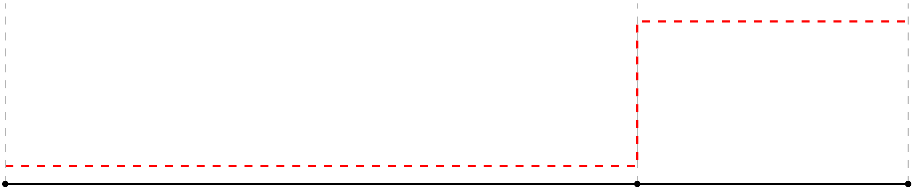
Exising solutions: limiting, adaptive refinement, artificial viscosity

usually result in order reduction or very fine discretizations

Proposed solution

align features of solution basis with features in the solution using
optimization formulation and solver

Optimization beyond design/control: high-order shock resolution



Fundamental issue: interpolate discontinuity with polynomial basis

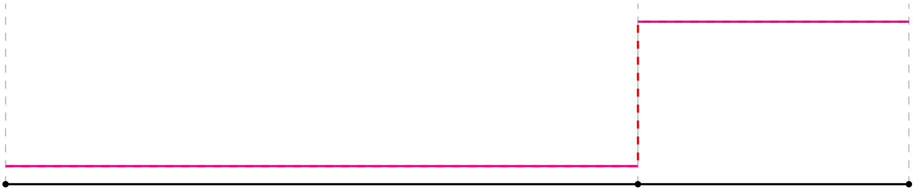
Exising solutions: limiting, adaptive refinement, artificial viscosity

usually result in order reduction or very fine discretizations

Proposed solution

align features of solution basis with features in the solution using
optimization formulation and solver

Optimization beyond design/control: high-order shock resolution



Fundamental issue: interpolate discontinuity with polynomial basis

Exising solutions: limiting, adaptive refinement, artificial viscosity

usually result in order reduction or very fine discretizations

Proposed solution

align features of solution basis with features in the solution using ***optimization formulation and solver***

Shock tracking optimization formulation

- Consider the spatial discretization of the conservation law

$$\nabla_{\boldsymbol{X}} \cdot \boldsymbol{F}(\boldsymbol{U};\, \boldsymbol{X}) = \boldsymbol{0} \qquad \rightarrow \qquad \boldsymbol{r}(\boldsymbol{u};\, \boldsymbol{x}) = \boldsymbol{0}$$

  - $\boldsymbol{U}$, $\boldsymbol{X}$ are the continuous state vector and coordinate
  - $\boldsymbol{x}$ contains the coordinates of the mesh nodes
  - $\boldsymbol{u}$ contains the discrete state vector corresponding to $\boldsymbol{U}$ at the mesh nodes

- Shock tracking formulation

$$\begin{aligned} \underset{\boldsymbol{u},\,\boldsymbol{x}}{\text{minimize}} \quad & f(\boldsymbol{u},\, \boldsymbol{x}) \\ \text{subject to} \quad & \boldsymbol{r}(\boldsymbol{u};\, \boldsymbol{x}) = 0 \end{aligned}$$

Key assumption: Solution basis supports discontinuties along element ***edges****, i.e., discontinuous Galerkin, finite volume*

Shock tracking objective function

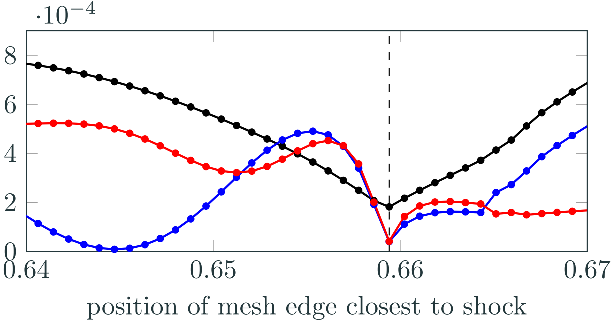
Requirements on objective function

obtains minimum *when mesh edge aligned with shock and* ***monotonically*** *decreases to minimum in (large) neighborhood*

$$f(\boldsymbol{u};\boldsymbol{x}) = f_{shk}(\boldsymbol{u};\boldsymbol{x}) + \alpha f_{msh}(\boldsymbol{x})$$

$$f_{shk}(\boldsymbol{u},\boldsymbol{x}) = \sum_{e=1}^{n_e} \int_{\Omega_e(\boldsymbol{x})} \left|\boldsymbol{u} - \bar{\boldsymbol{u}}\right|^2 \, dV$$

$$f_{msh}(\boldsymbol{x}) = \sum_{e=1}^{n_e} \sum_{k=1}^{n_q^e} \left| \frac{\mathrm{tr}\, \boldsymbol{G}^T \boldsymbol{G}}{\det \boldsymbol{G}} \right|$$



Objective function as an element edge is smoothly swept across shock location for: $f_{shk}(\boldsymbol{u},\boldsymbol{x})$ (black), residual-based objective (blue), and Rankine-Hugniot-based objective (red).

Full space optimization solver for shock tracking

Cannot use ***nested approach*** *to PDE optimization because it requires solving* $\boldsymbol{r}(\boldsymbol{u};\boldsymbol{x}) = 0$ *for* $\boldsymbol{x} \neq \boldsymbol{x}^* \implies$ ***crash***

- **Full space approach**: $\boldsymbol{u} \rightarrow \boldsymbol{u}^*$ and $\boldsymbol{x} \rightarrow \boldsymbol{x}^*$ *simultaneously*
- Define Lagrangian

$$\mathcal{L}(\boldsymbol{u}, \boldsymbol{x}, \boldsymbol{\lambda}) = f(\boldsymbol{u}; \boldsymbol{x}) - \boldsymbol{\lambda}^T \boldsymbol{r}(\boldsymbol{u}; \boldsymbol{x})$$

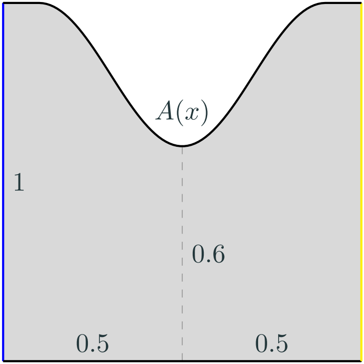
- First-order optimality (KKT) conditions for full space optimization problem

$$\nabla_{\boldsymbol{u}}\mathcal{L}(\boldsymbol{u}^*, \boldsymbol{x}^*, \boldsymbol{\lambda}^*) = \boldsymbol{0}, \qquad \nabla_{\boldsymbol{x}}\mathcal{L}(\boldsymbol{u}^*, \boldsymbol{x}^*, \boldsymbol{\lambda}^*) = \boldsymbol{0}, \qquad \nabla_{\boldsymbol{\lambda}}\mathcal{L}(\boldsymbol{u}^*, \boldsymbol{x}^*, \boldsymbol{\lambda}^*) = \boldsymbol{0}$$

- Apply (quasi-)Newton method[3] to solve nonlinear KKT system for $\boldsymbol{u}^*, \boldsymbol{x}^*, \boldsymbol{\lambda}^*$

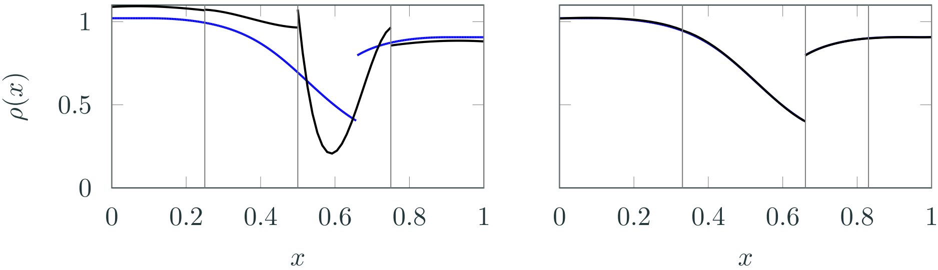
[3] usually requires globalization such as linesearch or trust-region

Nozzle flow: quasi-1d Euler equations



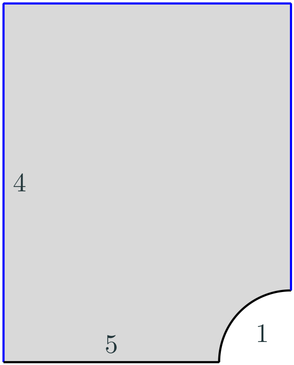
Geometry and boundary conditions for nozzle flow. Boundary conditions: inviscid wall (——), inflow (——), outflow (——).

Resolution of 1d transonic flow with only 4 *quartic elements*



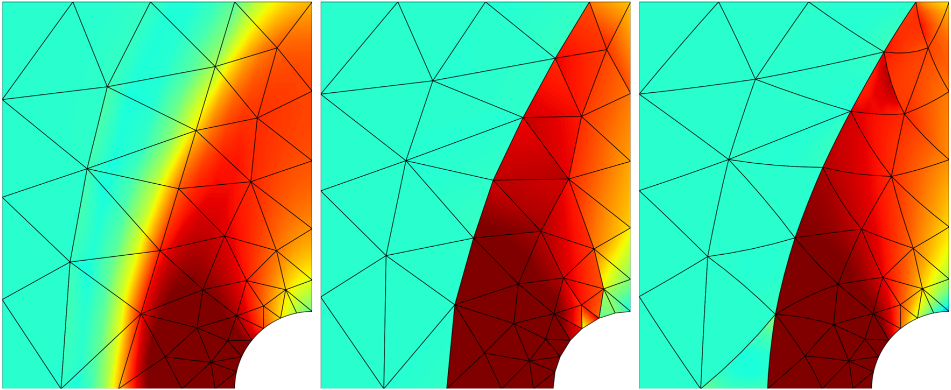
The solution of the quasi-1d Euler equations using: 300 linear elements (——) and 4 quartic elements (——) on a mesh not aligned (*left*) and aligned (*right*) with the shock.

Supersonic flow around cylinder: 2D Euler equations



Geometry and boundary conditions for supersonic flow around cylinder. Boundary conditions: inviscid wall/symmetry condition (——) and farfield (——).

Resolution of 2D supersonic flow with only 67 *quadratic elements*

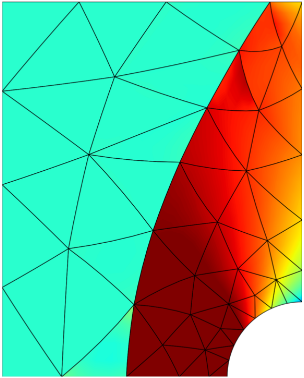


The solution of the 2d Euler equations using: 67 quadratic elements on a mesh not aligned with the shock (*left*), 67 linear elements on a mesh aligned with the shock (*middle*), 67 quadratic elements on a mesh aligned with the shock (*right*).

Convergence to optimal solution and mesh

PDE-constrained optimization for design/control and beyond

- Globally high-order numerical method and adjoint-based gradient computations for efficient **design and data assimilation**
  - energetically optimal flapping, energy harvesting mechanisms, super-resolution MRI
- Globally convergent multifidelity framework for PDE-constrained **optimization under uncertainty**
  - risk-averse flow control
- Optimization-based **shock tracking framework** for highly resolved supersonic flows on extremely coarse meshes



References I

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.

Kouri, D. P., Heinkenschloss, M., Ridzal, D., and van Bloemen Waanders, B. G. (2014).
Inexact objective function evaluations in a trust-region algorithm for PDE-constrained optimization under uncertainty.
SIAM Journal on Scientific Computing, 36(6):A3011–A3029.

Wang, J., Zahr, M. J., and Persson, P.-O. (6/5/2017 – 6/9/2017).
Energetically optimal flapping flight based on a fully discrete adjoint method with explicit treatment of flapping frequency.
In *Proc. of the 23rd AIAA Computational Fluid Dynamics Conference*, Denver, Colorado. American Institute of Aeronautics and Astronautics.

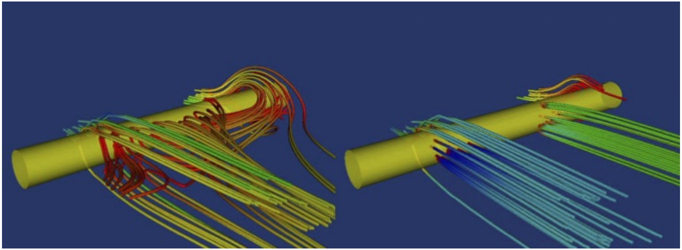
Zahr, M. J., Huang, Z., and Persson, P.-O. (1/8/2018 – 1/12/2018).
Adjoint-based optimization of time-dependent fluid-structure systems using a high-order discontinuous Galerkin discretization.
In *AIAA Science and Technology Forum and Exposition (SciTech2018)*, Kissimmee, Florida. American Institute of Aeronautics and Astronautics.

Zahr, M. J. and Persson, P.-O. (2016).
An adjoint method for a high-order discretization of deforming domain conservation laws for optimization of flow problems.
Journal of Computational Physics.

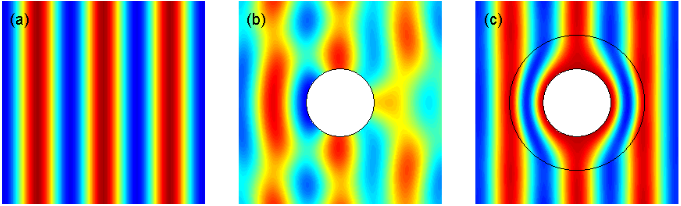
Zahr, M. J., Persson, P.-O., and Wilkening, J. (2016).
A fully discrete adjoint method for optimization of flow problems on deforming domains with time-periodicity constraints.
Computers & Fluids.

PDE optimization is ubiquitous in science and engineering

Control: Drive system to a desired state



Boundary flow control



Metamaterial cloaking – electromagnetic invisibility

High-order discretization of PDE-constrained optimization

- *Continuous* PDE-constrained optimization problem

$$\begin{aligned} \underset{\boldsymbol{U},\ \boldsymbol{\mu}}{\text{minimize}} \quad & \mathcal{J}(\boldsymbol{U}, \boldsymbol{\mu}) \\ \text{subject to} \quad & \boldsymbol{C}(\boldsymbol{U}, \boldsymbol{\mu}) \leq 0 \\ & \frac{\partial \boldsymbol{U}}{\partial t} + \nabla \cdot \boldsymbol{F}(\boldsymbol{U}, \nabla \boldsymbol{U}) = 0 \quad \text{in} \quad v(\boldsymbol{\mu}, t) \end{aligned}$$

- *Fully discrete* PDE-constrained optimization problem

$$\begin{aligned} \underset{\substack{\boldsymbol{u}_0,\ \dots,\ \boldsymbol{u}_{N_t} \in \mathbb{R}^{N_{\boldsymbol{u}}}, \\ \boldsymbol{k}_{1,1},\ \dots,\ \boldsymbol{k}_{N_t,s} \in \mathbb{R}^{N_{\boldsymbol{u}}}, \\ \boldsymbol{\mu} \in \mathbb{R}^{n_{\boldsymbol{\mu}}}}}{\text{minimize}} \quad & J(\boldsymbol{u}_0,\ \dots,\ \boldsymbol{u}_{N_t},\ \boldsymbol{k}_{1,1},\ \dots,\ \boldsymbol{k}_{N_t,s},\ \boldsymbol{\mu}) \\ \text{subject to} \quad & \mathbf{C}(\boldsymbol{u}_0,\ \dots,\ \boldsymbol{u}_{N_t},\ \boldsymbol{k}_{1,1},\ \dots,\ \boldsymbol{k}_{N_t,s},\ \boldsymbol{\mu}) \leq 0 \\ & \boldsymbol{u}_0 - \boldsymbol{g}(\boldsymbol{\mu}) = 0 \\ & \boldsymbol{u}_n - \boldsymbol{u}_{n-1} - \sum_{i=1}^{s} b_i \boldsymbol{k}_{n,i} = 0 \\ & \boldsymbol{M}\boldsymbol{k}_{n,i} - \Delta t_n \boldsymbol{r}\left(\boldsymbol{u}_{n,i},\ \boldsymbol{\mu},\ t_{n,i}\right) = 0 \end{aligned}$$

BERKELEY LAB

Adjoint equation derivation: outline

- Define **auxiliary** PDE-constrained optimization problem

$$
\begin{aligned}
\underset{\substack{\boldsymbol{u}_0,\ \ldots,\ \boldsymbol{u}_{N_t} \in \mathbb{R}^{N_{\boldsymbol{u}}},\\ \boldsymbol{k}_{1,1},\ \ldots,\ \boldsymbol{k}_{N_t,s} \in \mathbb{R}^{N_{\boldsymbol{u}}}}}{\text{minimize}} \quad & F(\boldsymbol{u}_0,\ \ldots,\ \boldsymbol{u}_{N_t},\ \boldsymbol{k}_{1,1},\ \ldots,\ \boldsymbol{k}_{N_t,s},\ \boldsymbol{\mu}) \\
\text{subject to} \quad & \boldsymbol{R}_0 = \boldsymbol{u}_0 - \boldsymbol{g}(\boldsymbol{\mu}) = 0 \\
& \boldsymbol{R}_n = \boldsymbol{u}_n - \boldsymbol{u}_{n-1} - \sum_{i=1}^{s} b_i \boldsymbol{k}_{n,i} = 0 \\
& \boldsymbol{R}_{n,i} = \boldsymbol{M}\boldsymbol{k}_{n,i} - \Delta t_n \boldsymbol{r}\left(\boldsymbol{u}_{n,i},\ \boldsymbol{\mu},\ t_{n,i}\right) = 0
\end{aligned}
$$

- Define **Lagrangian**

$$
\mathcal{L}(\boldsymbol{u}_n,\ \boldsymbol{k}_{n,i},\ \boldsymbol{\lambda}_n,\ \boldsymbol{\kappa}_{n,i}) = F - {\boldsymbol{\lambda}_0}^T \boldsymbol{R}_0 - \sum_{n=1}^{N_t} {\boldsymbol{\lambda}_n}^T \boldsymbol{R}_n - \sum_{n=1}^{N_t} \sum_{i=1}^{s} {\boldsymbol{\kappa}_{n,i}}^T \boldsymbol{R}_{n,i}
$$

- The solution of the optimization problem is given by the **Karush-Kuhn-Tucker (KKT) sytem**

$$
\frac{\partial \mathcal{L}}{\partial \boldsymbol{u}_n} = 0, \quad \frac{\partial \mathcal{L}}{\partial \boldsymbol{k}_{n,i}} = 0, \quad \frac{\partial \mathcal{L}}{\partial \boldsymbol{\lambda}_n} = 0, \quad \frac{\partial \mathcal{L}}{\partial \boldsymbol{\kappa}_{n,i}} = 0
$$

BERKELEY LAB

Extension: constraint requiring time-periodicity [Zahr et al., 2016]

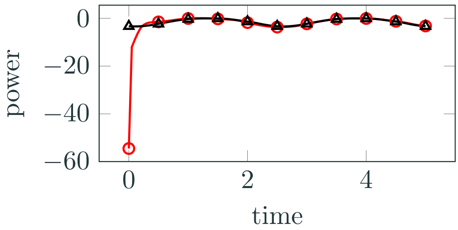
- Optimization of *cyclic* problems requires finding time-periodic solution of PDE
- Necessary for physical relevance and avoid transients that may lead to crash

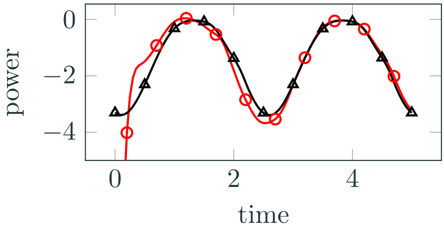
$$\begin{aligned} \underset{\boldsymbol{U},\ \boldsymbol{\mu}}{\text{minimize}} \quad & \mathcal{F}(\boldsymbol{U}, \boldsymbol{\mu}) \\ \text{subject to} \quad & \boldsymbol{U}(\boldsymbol{x}, 0) = \boldsymbol{U}(\boldsymbol{x}, T) \\ & \frac{\partial \boldsymbol{U}}{\partial t} + \nabla \cdot \boldsymbol{F}(\boldsymbol{U}, \nabla \boldsymbol{U}) = 0 \end{aligned}$$

$$\boldsymbol{\lambda}_{N_t} = \boldsymbol{\lambda}_0 + \frac{\partial F}{\partial \boldsymbol{u}_{N_t}}^T$$

$$\boldsymbol{\lambda}_{n-1} = \boldsymbol{\lambda}_n + \frac{\partial F}{\partial \boldsymbol{u}_{n-1}}^T + \sum_{i=1}^{s} \Delta t_n \frac{\partial \boldsymbol{r}_{n,i}}{\partial \boldsymbol{u}}^T \boldsymbol{\kappa}_{n,i}$$

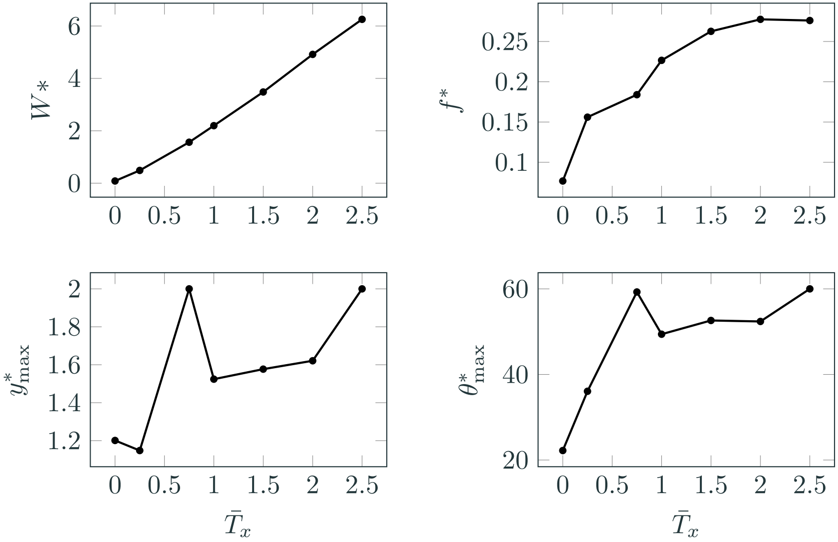
$$\boldsymbol{M}^T \boldsymbol{\kappa}_{n,i} = \frac{\partial F}{\partial \boldsymbol{u}_{N_t}}^T + b_i \boldsymbol{\lambda}_n + \sum_{j=i}^{s} a_{ji} \Delta t_n \frac{\partial \boldsymbol{r}_{n,i}}{\partial \boldsymbol{u}}^T \boldsymbol{\kappa}_{n,j}$$





Time history of power on airfoil of flow initialized from steady-state (red circle line) and from a time-periodic solution (black triangle line)

Energetically optimal flapping vs. required thrust: QoI



The optimal flapping energy (W^*), frequency (f^*), maximum heaving amplitude ($y^*_{\max}$), and maximum pitching amplitude ($\theta^*_{\max}$) as a function of the thrust constraint $\bar{T}_x$.

Extension: Multiphysics problems [Zahr et al., 2018]

- For problems that involve the interaction of multiple types of physical phenomena, *no changes required* if monolithic system considered

$$\begin{aligned} \boldsymbol{M}_0 \dot{\boldsymbol{u}}_0 &= \boldsymbol{r}_0(\boldsymbol{u}_0, \boldsymbol{c}_0(\boldsymbol{u}_0, \boldsymbol{u}_1)) \\ \boldsymbol{M}_1 \dot{\boldsymbol{u}}_1 &= \boldsymbol{r}_1(\boldsymbol{u}_1, \boldsymbol{c}_1(\boldsymbol{u}_0, \boldsymbol{u}_1)) \end{aligned}$$

- However, to solve in partitioned manner and achieve high-order, split as follows and apply implicit-explicit Runge-Kutta

$$\begin{aligned} \boldsymbol{M}_0 \dot{\boldsymbol{u}}_0 &= \boldsymbol{r}_0(\boldsymbol{u}_0, \boldsymbol{c}_0(\boldsymbol{u}_0, \boldsymbol{u}_1)) \\ \boldsymbol{M}_1 \dot{\boldsymbol{u}}_1 &= \boldsymbol{r}_1(\boldsymbol{u}_1, \tilde{\boldsymbol{c}}_1) \qquad + \left(\boldsymbol{r}_1(\boldsymbol{u}_1, \boldsymbol{c}_1(\boldsymbol{u}_0, \boldsymbol{u}_1)) - \boldsymbol{r}_1(\boldsymbol{u}_1, \tilde{\boldsymbol{c}}_1)\right) \end{aligned}$$

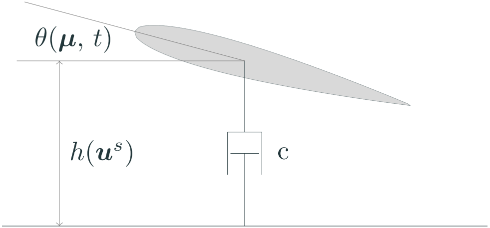
- Adjoint equations inherit explicit-implicit structure

Optimal energy harvesting from foil-damper system

Goal: Maximize energy harvested from foil-damper system

$$\underset{\boldsymbol{\mu}}{\text{maximize}} \quad \frac{1}{T}\int_0^T (c\dot{h}^2(\boldsymbol{u}^s) - M_z(\boldsymbol{u}^f)\dot{\theta}(\boldsymbol{\mu},\, t))\, dt$$

- Fluid: Isentropic Navier-Stokes on deforming domain (ALE)
- Structure: Force balance in y-direction between foil and damper
- Motion driven by *imposed* $\theta(\boldsymbol{\mu},\, t) = \mu_1 \cos(2\pi f t)$



$\mu_1^* \approx 45^\circ$

MRI data assimilation formulation

- $\boldsymbol{d}^*_{i,n}$: MRI measurement taken in voxel i at the nth time sample
- $\boldsymbol{d}_{i,n}(\boldsymbol{U},\, \boldsymbol{\mu})$: computational representation of $\boldsymbol{d}^*_{i,n}$

$$
\begin{aligned}
\boldsymbol{d}_{i,n}(\boldsymbol{U},\, \boldsymbol{\mu}) &= \int_0^T \int_V w_{i,n}(\boldsymbol{x},\, t) \cdot \boldsymbol{U}(\boldsymbol{x},\, t)\, dV\, dt \\
w_{i,n}(\boldsymbol{x},\, t) &= \chi_s(\boldsymbol{x};\, \boldsymbol{x}_i,\, \Delta\boldsymbol{x})\chi_t(t;\, t_n,\, \Delta t) \\
\chi_t(s;\, c,\, w) &= \frac{1}{1 + e^{-(s-(c-0.5w))/\sigma}} - \frac{1}{1 + e^{-(s-(c+0.5w))/\sigma}} \\
\chi_s(\boldsymbol{x};\, \boldsymbol{c},\, \boldsymbol{w}) &= \chi_t(x_1;\, c_1,\, w_1)\chi_t(x_2;\, c_2,\, w_2)\chi_t(x_3;\, c_3,\, w_3)
\end{aligned}
$$

- $\boldsymbol{x}_i$ - center of ith MRI voxel
- t_n time instance of n MRI sample
- $\Delta\boldsymbol{x}$ - size of MRI voxel in each dimension
- Δt sampling interval in time

$$
\boxed{\underset{\boldsymbol{U},\, \boldsymbol{\mu}}{\text{minimize}} \quad \sum_{i=1}^{n_{xyz}} \sum_{n=1}^{n_t} \frac{\alpha_{i,n}}{2} \left|\left| \boldsymbol{d}_{i,n}(\boldsymbol{U},\, \boldsymbol{\mu}) - \boldsymbol{d}^*_{i,n} \right|\right|_2^2}
$$

BERKELEY LAB

Coarse MRI grid (24×36), 20 time samples, 3% noise

Reconstructed flow

Synthetic MRI data $\boldsymbol{d}^*_{i,n}$ (top) and computational representation of MRI data $\boldsymbol{d}_{i,n}$ (bottom)

Fine MRI grid (40×60), 20 time samples, 3% noise

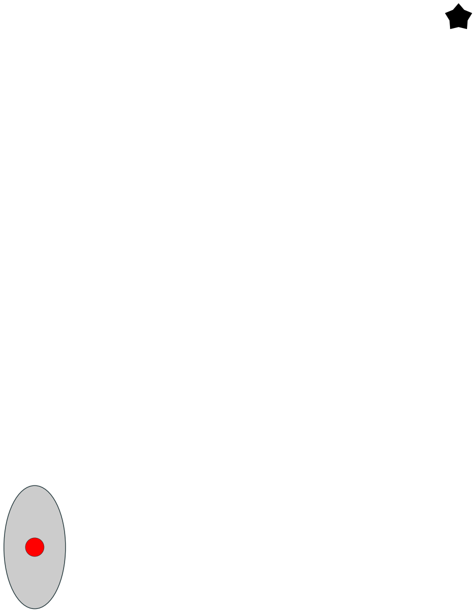
Reconstructed flow

Synthetic MRI data $\boldsymbol{d}^*_{i,n}$ (top) and computational representation of MRI data $\boldsymbol{d}_{i,n}$ (bottom)

Trust region framework for optimization with ROMs

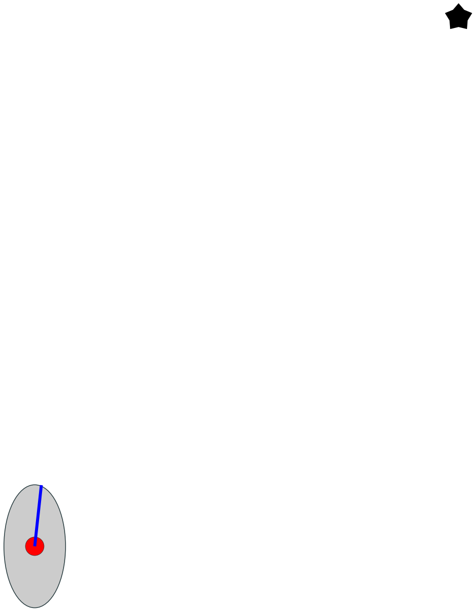
$\boldsymbol{\mu}$-space

Trust region framework for optimization with ROMs

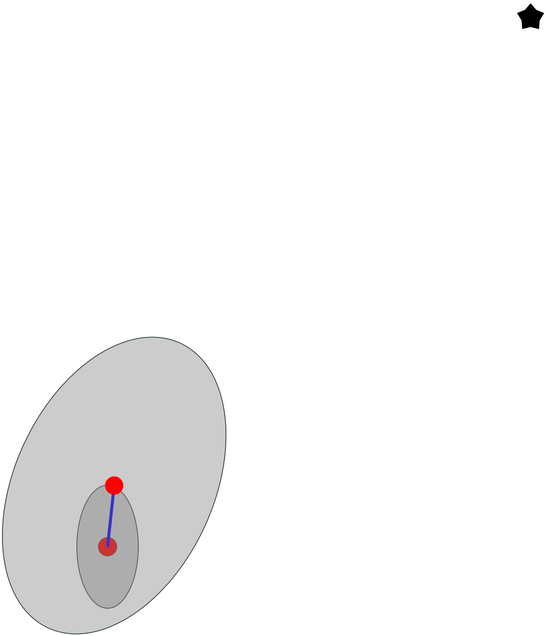


$\boldsymbol{\mu}$-space

Trust region framework for optimization with ROMs

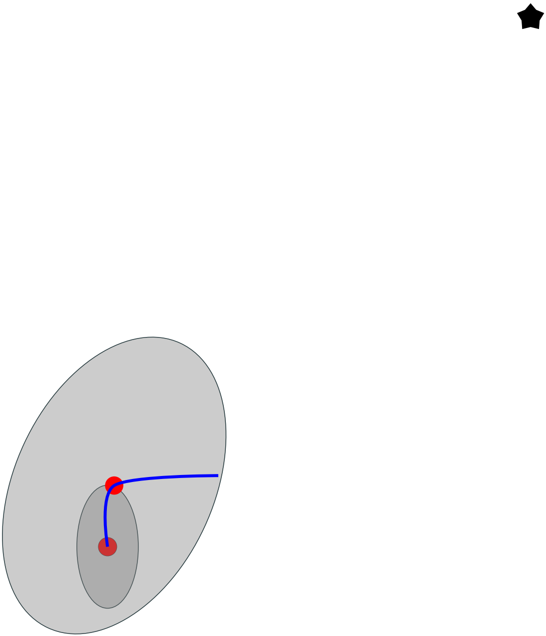


Trust region framework for optimization with ROMs



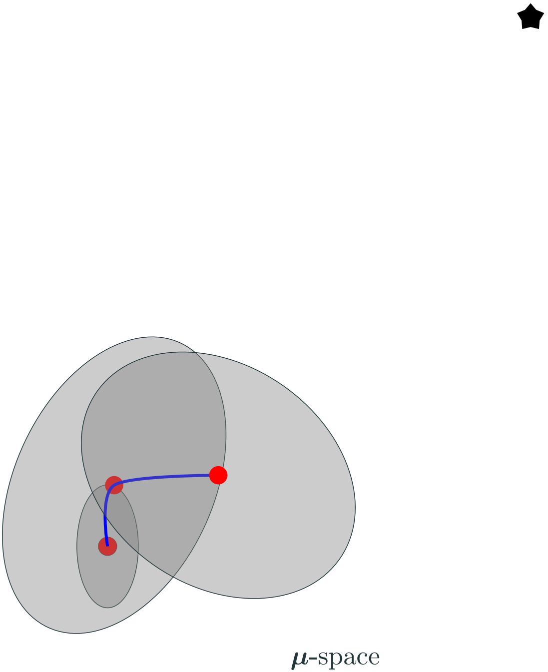
$\boldsymbol{\mu}$-space

Trust region framework for optimization with ROMs

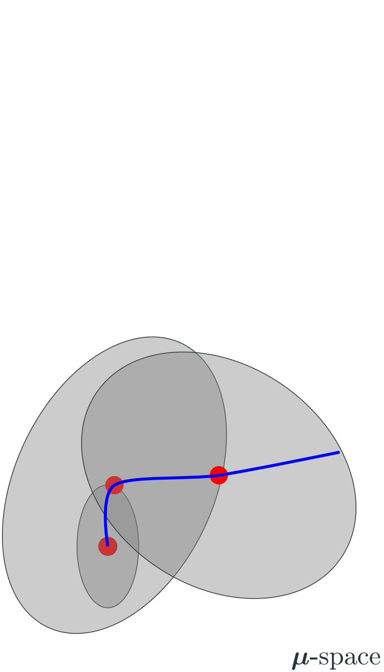


μ-space

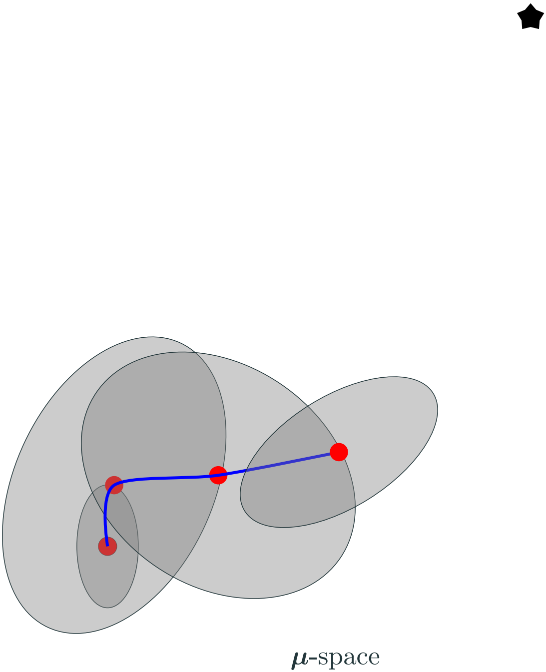
Trust region framework for optimization with ROMs



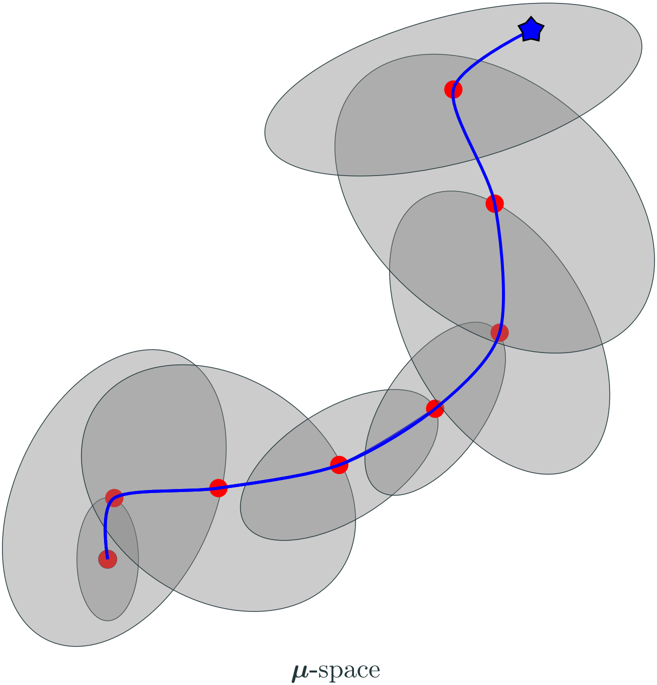
Trust region framework for optimization with ROMs



Trust region framework for optimization with ROMs



Trust region framework for optimization with ROMs



Trust region ingredients for global convergence

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

Approximation models

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

Error indicators

$$\begin{aligned} ||\nabla F(\boldsymbol{\mu}) - \nabla m_k(\boldsymbol{\mu})|| &\leq \xi \varphi_k(\boldsymbol{\mu}) \qquad && \xi > 0 \\ |F(\boldsymbol{\mu}_k) - F(\boldsymbol{\mu}) + \psi_k(\boldsymbol{\mu}) - \psi_k(\boldsymbol{\mu}_k)| &\leq \sigma \theta_k(\boldsymbol{\mu}) \qquad && \sigma > 0 \end{aligned}$$

Adaptivity

$$\begin{aligned} \varphi_k(\boldsymbol{\mu}_k) &\leq \kappa_\varphi \min\{||\nabla m_k(\boldsymbol{\mu}_k)||\,,\ \Delta_k\} \\ \theta_k(\hat{\boldsymbol{\mu}}_k)^\omega &\leq \eta \min\{m_k(\boldsymbol{\mu}_k) - m_k(\hat{\boldsymbol{\mu}}_k),\ r_k\} \end{aligned}$$

BERKELEY LAB

Trust region method with inexact gradients and objective

1: **Model update**: Choose model m_k and error indicator φ_k

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

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 approximation of actual-to-predicted reduction

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

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

Final requirement for convergence: Adaptivity

With the approximation model, $m_k(\boldsymbol{\mu})$, and gradient error indicator, $\varphi_k(\boldsymbol{\mu})$

$$m_k(\boldsymbol{\mu}) = \mathbb{E}_{\mathcal{I}_k}\left[\mathcal{J}(\boldsymbol{\Phi}_k \boldsymbol{u}_r(\boldsymbol{\mu}, \cdot), \boldsymbol{\mu}, \cdot)\right]$$
$$\varphi_k(\boldsymbol{\mu}) = \alpha_1 \mathcal{E}_1(\boldsymbol{\mu}; \mathcal{I}_k, \boldsymbol{\Phi}_k) + \alpha_2 \mathcal{E}_2(\boldsymbol{\mu}; \mathcal{I}_k, \boldsymbol{\Phi}_k) + \alpha_3 \mathcal{E}_4(\boldsymbol{\mu}; \mathcal{I}_k, \boldsymbol{\Phi}_k)$$

the sparse grid $\mathcal{I}_k$ and reduced-order basis $\boldsymbol{\Phi}_k$ must be constructed such that the gradient condition holds

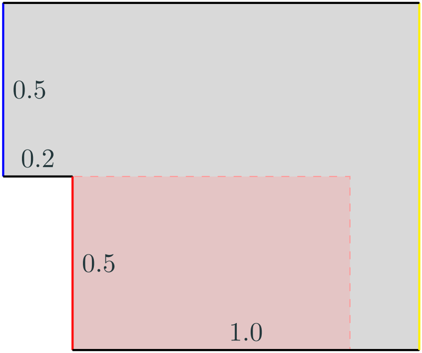
$$\varphi_k(\boldsymbol{\mu}_k) \leq \kappa_\varphi \min\{||\nabla m_k(\boldsymbol{\mu}_k)||\,,\, \Delta_k\}$$

Define dimension-adaptive greedy method to target each source of error such that the stronger conditions hold

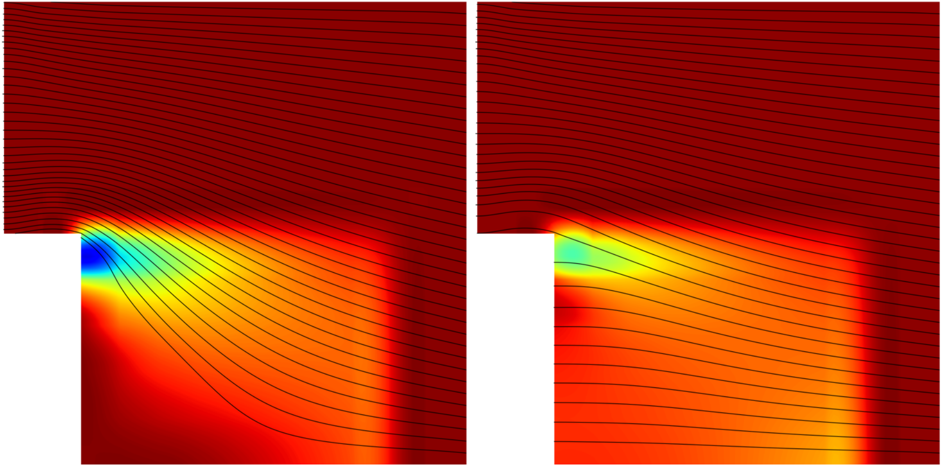
$$\mathcal{E}_1(\boldsymbol{\mu}_k; \mathcal{I}, \boldsymbol{\Phi}) \leq \frac{\kappa_\varphi}{3\alpha_1} \min\{||\nabla m_k(\boldsymbol{\mu}_k)||\,,\, \Delta_k\}$$
$$\mathcal{E}_2(\boldsymbol{\mu}_k; \mathcal{I}, \boldsymbol{\Phi}) \leq \frac{\kappa_\varphi}{3\alpha_2} \min\{||\nabla m_k(\boldsymbol{\mu}_k)||\,,\, \Delta_k\}$$
$$\mathcal{E}_4(\boldsymbol{\mu}_k; \mathcal{I}, \boldsymbol{\Phi}) \leq \frac{\kappa_\varphi}{3\alpha_3} \min\{||\nabla m_k(\boldsymbol{\mu}_k)||\,,\, \Delta_k\}$$

BERKELEY LAB

Backward facing step: minimize recirculation



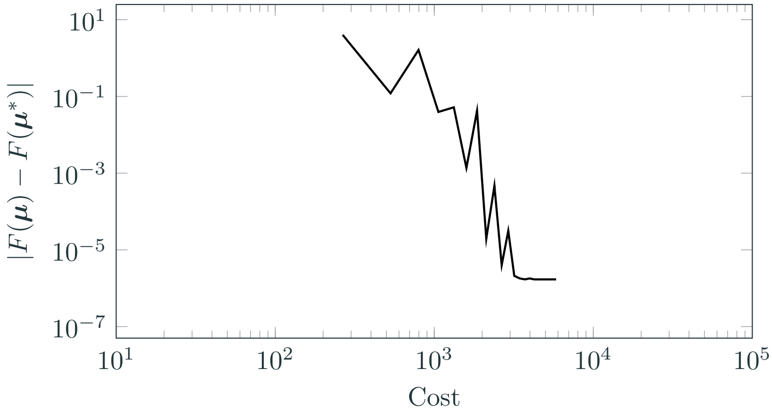
Geometry and boundary conditions for backward facing step. Boundary conditions: viscous wall (——), parametrized inflow($\boldsymbol{\mu}$) (——), stochastic inflow($\boldsymbol{\xi}$) (——), outflow (——). Vorticity magnitude minimized in red shaded region.



Mean vorticity corresponding to no inflow (left) and optimal inflow (right) along parametrized boundary.

Significant reduction in cost, if ROM only $10\times$ faster than HDM

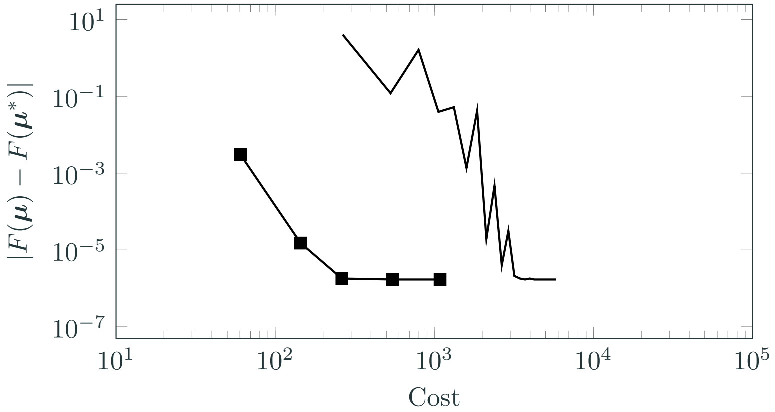
$$\text{Cost} = \text{nHdmPrim} + 0.5 \times \text{nHdmAdj} + \tau^{-1} \times (\text{nRomPrim} + 0.5 \times \text{nRomAdj})$$



5-level isotropic SG (——), dimension-adaptive SG [Kouri et al., 2014] ( ), and proposed ROM/SG for $\tau = 1$ ( ), $\tau = 10$ ( ), $\tau = 100$ ( ), $\tau = \infty$ ( )

Significant reduction in cost, if ROM only $10\times$ faster than HDM

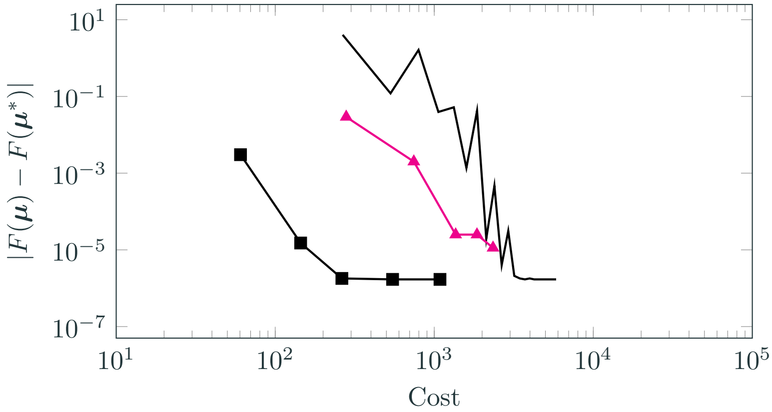
$$\text{Cost} = \text{nHdmPrim} + 0.5 \times \text{nHdmAdj} + \tau^{-1} \times (\text{nRomPrim} + 0.5 \times \text{nRomAdj})$$



5-level isotropic SG (——), dimension-adaptive SG [Kouri et al., 2014] (—■—), and proposed ROM/SG for $\tau = 1$ ( ), $\tau = 10$ ( ), $\tau = 100$ ( ), $\tau = \infty$ ( )

Significant reduction in cost, if ROM only $10\times$ faster than HDM

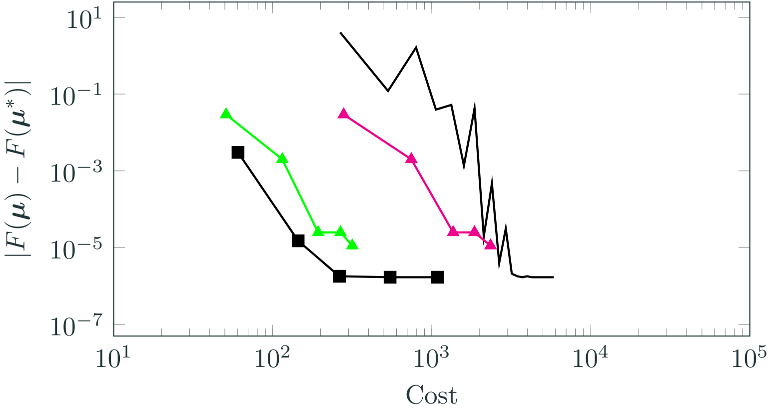
$$\text{Cost} = \text{nHdmPrim} + 0.5 \times \text{nHdmAdj} + \tau^{-1} \times (\text{nRomPrim} + 0.5 \times \text{nRomAdj})$$



5-level isotropic SG (——), dimension-adaptive SG [Kouri et al., 2014] (—■—), and proposed ROM/SG for $\tau = 1$ (—▲—), $\tau = 10$ ( ), $\tau = 100$ ( ), $\tau = \infty$ ( )

Significant reduction in cost, if ROM only $10\times$ faster than HDM

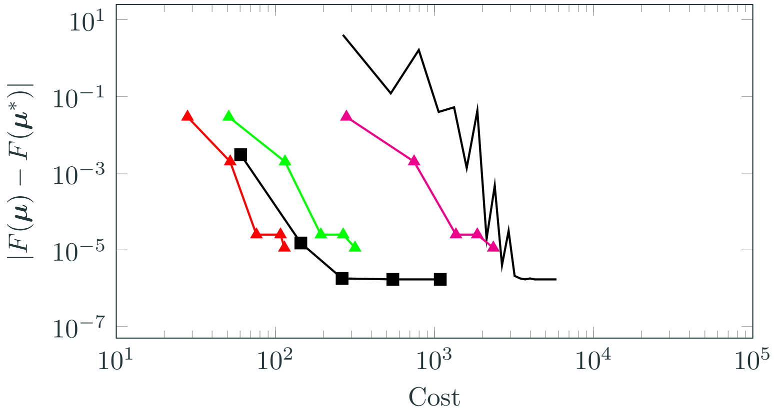
$$\text{Cost} = \text{nHdmPrim} + 0.5 \times \text{nHdmAdj} + \tau^{-1} \times (\text{nRomPrim} + 0.5 \times \text{nRomAdj})$$



5-level isotropic SG (——), dimension-adaptive SG [Kouri et al., 2014] (—■—), and proposed ROM/SG for $\tau = 1$ (—▲—), $\tau = 10$ (—▲—), $\tau = 100$ ( ), $\tau = \infty$ ( )

Significant reduction in cost, if ROM only $10\times$ faster than HDM

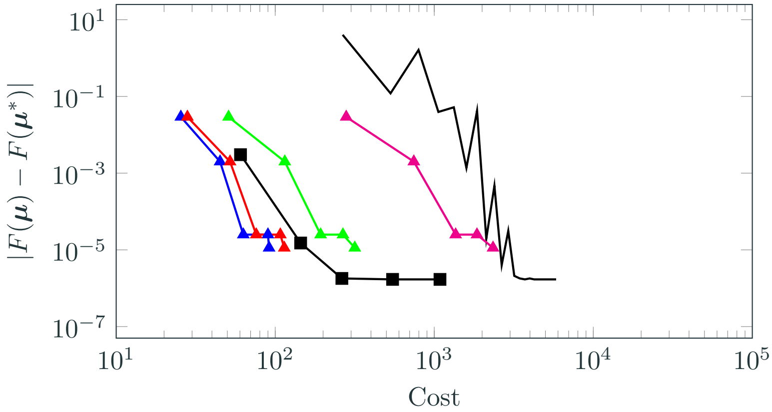
$$\text{Cost} = \text{nHdmPrim} + 0.5 \times \text{nHdmAdj} + \tau^{-1} \times (\text{nRomPrim} + 0.5 \times \text{nRomAdj})$$



5-level isotropic SG (——), dimension-adaptive SG [Kouri et al., 2014] (—■—), and proposed ROM/SG for $\tau = 1$ (—▲—), $\tau = 10$ (—▲—), $\tau = 100$ (—▲—), $\tau = \infty$ ( )

Significant reduction in cost, if ROM only $10\times$ faster than HDM

$$\text{Cost} = \text{nHdmPrim} + 0.5 \times \text{nHdmAdj} + \tau^{-1} \times (\text{nRomPrim} + 0.5 \times \text{nRomAdj})$$



5-level isotropic SG (—), dimension-adaptive SG [Kouri et al., 2014] (–■–), and proposed ROM/SG for $\tau = 1$ (–▲–), $\tau = 10$ (–▲–), $\tau = 100$ (–▲–), $\tau = \infty$ (–▲–)