# Modeling and controlling reduced-order-model uncertainty in data assimilation

Kevin Carlberg

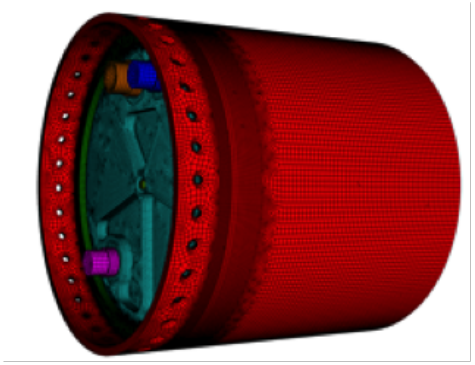
Sandia National Laboratories

Livermore, California

SUPRI-B group seminar
Stanford, CA
November 17, 2014

*Modeling and simulation*

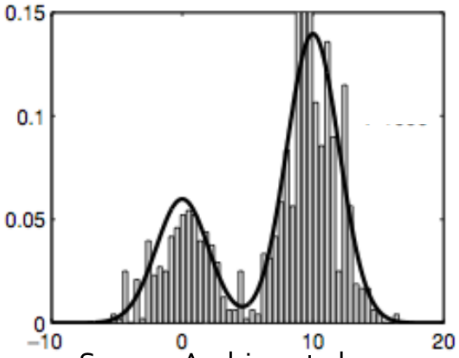
+ Indispensible tool
- Only a few simulations possible



## Barrier

*Time-critical applications*

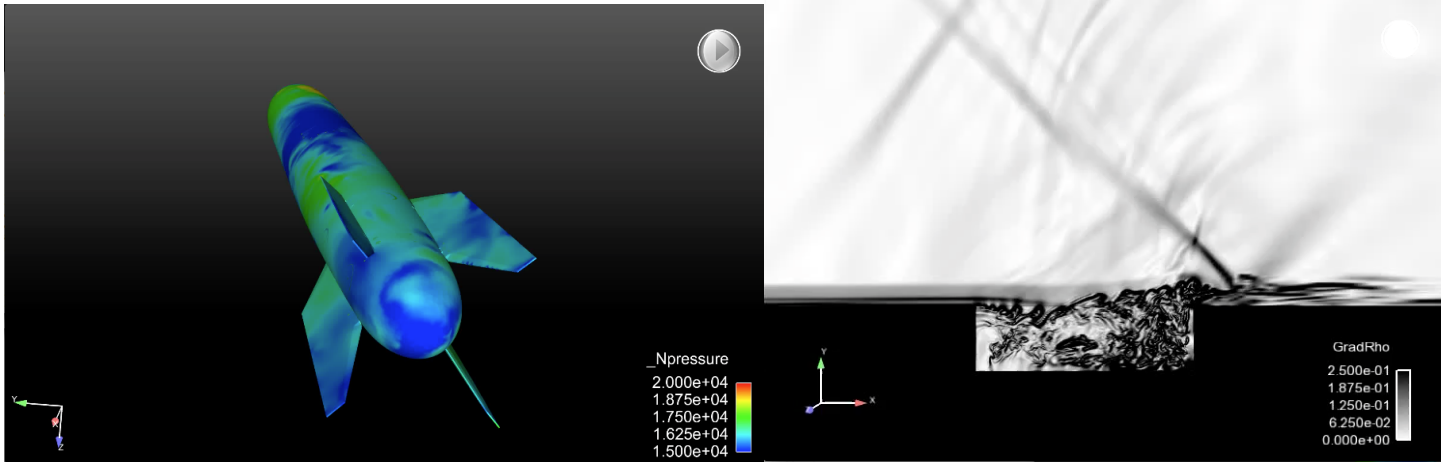
- Many query
- Real time



Source: Andrieu et al., 2003

**Objective: break barrier**

# At Sandia




- Full-order model
  - 100 million cells
  - 200,000 time steps
- High simulation costs
  - 6 weeks
  - 5000 cores
  - 6 runs maxes out Cielo

## Barrier

- Design engineers require faster simulations
- Uncertainty quantification (UQ)

# Data assimilation

source: Holger Speckmann, Airbus

*Given sensor data, what is the updated knowledge of material properties?*

- Bayesian inference problem

# Bayesian inference

$$\text{inputs } \boldsymbol{\mu} \rightarrow \boxed{\text{full-order model (FOM)}} \rightarrow \text{outputs } \mathbf{z}$$

$$P(\boldsymbol{\mu}|\bar{\mathbf{z}}) \propto P(\bar{\mathbf{z}}|\boldsymbol{\mu})P(\boldsymbol{\mu})$$

**Given**:

- Prior: $P(\boldsymbol{\mu})$
- Measurements: $\bar{\mathbf{z}}$
- Measurements/FOM relationship: $\bar{\mathbf{z}} = \mathbf{z}(\boldsymbol{\mu}^\star) + \boldsymbol{\varepsilon}$
  - Sensor noise: $\boldsymbol{\varepsilon} \sim \mathcal{N}(0, \sigma^2\mathbf{I})$
  - Likelihood: $P(\bar{\mathbf{z}}|\boldsymbol{\mu}) \sim \mathcal{N}(\mathbf{z}(\boldsymbol{\mu}), \sigma^2\mathbf{I})$

**Compute**:

- Samples of the posterior $P(\boldsymbol{\mu}|\bar{\mathbf{z}})$

- Expensive: Likelihood samples requires FOM evaluations

# Bayesian inference with *reduced-order models*

$$\text{inputs } \boldsymbol{\mu} \rightarrow \boxed{\text{reduced-order model (ROM)}} \rightarrow \text{outputs } \mathbf{z}_{\text{red}}$$

$$P(\boldsymbol{\mu}|\bar{\mathbf{z}}) \propto P(\bar{\mathbf{z}}|\boldsymbol{\mu})P(\boldsymbol{\mu})$$

**Given**:

- Prior: $P(\boldsymbol{\mu})$
- Measurements: $\bar{\mathbf{z}}$
- Measurements/ROM relationship: $\bar{\mathbf{z}} = \mathbf{z}_{\text{red}}(\boldsymbol{\mu}^\star) + \boldsymbol{\varepsilon}$
  - Sensor noise: $\boldsymbol{\varepsilon} \sim \mathcal{N}(0, \sigma^2\mathbf{I})$
  - Likelihood: $P(\bar{\mathbf{z}}|\boldsymbol{\mu}) \sim \mathcal{N}(\mathbf{z}_{\text{red}}(\boldsymbol{\mu}), \sigma^2\mathbf{I})$

**Compute**:

- Samples of the posterior $P(\boldsymbol{\mu}|\bar{\mathbf{z}})$

\+ Cheap: Likelihood samples requires ROM evaluations

\- Neglects ROM errors:

$$\begin{aligned}\bar{\mathbf{z}} &= \mathbf{z}(\boldsymbol{\mu}^\star) + \boldsymbol{\varepsilon} \\ &= \mathbf{z}_{\text{red}}(\boldsymbol{\mu}^\star) + \boxed{\delta_{\mathbf{z}}(\boldsymbol{\mu}^\star)} + \boldsymbol{\varepsilon}\end{aligned}$$

# Reduced-order modeling: modeling and controlling error

$$\bar{\mathbf{z}} = \mathbf{z}_{\text{red}}(\boldsymbol{\mu}^\star) + \boxed{\delta_{\mathbf{z}}(\boldsymbol{\mu}^\star)} + \varepsilon$$

*"An interesting future research direction is the inclusion of estimates of reduced model error as an additional source of uncertainty in the Bayesian formulation."*

*[Galbally et al., 2009]*

**Our goals**:

1. construct a statistical model of this error
2. control this error

# Error control

$$\bar{\mathbf{z}} = \mathbf{z}_{\text{red}}(\boldsymbol{\mu}^\star) + \boxed{\delta_{\mathbf{z}}(\boldsymbol{\mu}^\star)} + \varepsilon$$

*"An interesting future research direction is the inclusion of estimates of reduced model error as an additional source of uncertainty in the Bayesian formulation."*

*[Galbally et al., 2009]*

**Our goals**:

1. construct a statistical model of this error
2. control this error

# Reduced-order modeling: accuracy limitation

- FOM: parameterized sequence of systems of equations

$$\mathbf{r}^k(\mathbf{x}^k; \boldsymbol{\mu}) = 0, \quad k = 1, \ldots, t$$

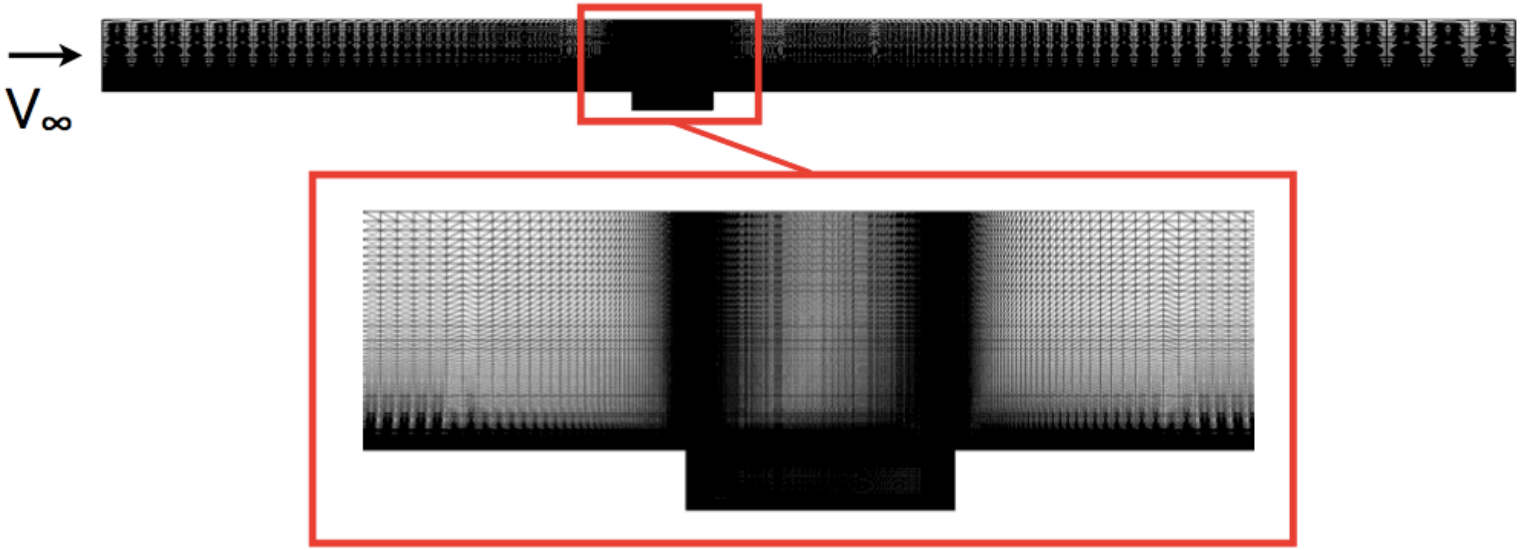
  with state $\mathbf{x} \in \mathbb{R}^n$, and residual $\mathbf{r}$.
- ROM
  1. *Offline*: construct low-dimensional basis $\mathbf{V} \in \mathbb{R}^{n \times p}$ with $p \ll n$
  2. *Online*: approximate $\mathbf{x}^k \approx \mathbf{V}\hat{\mathbf{x}}^k$ and solve

$$\mathbf{V}^T \mathbf{r}^k(\mathbf{V}\hat{\mathbf{x}}^k; \boldsymbol{\mu}) = 0, \quad k = 1, \ldots, t$$

  Additional approximations needed if $\mathbf{r}$ nonlinear or nonaffine

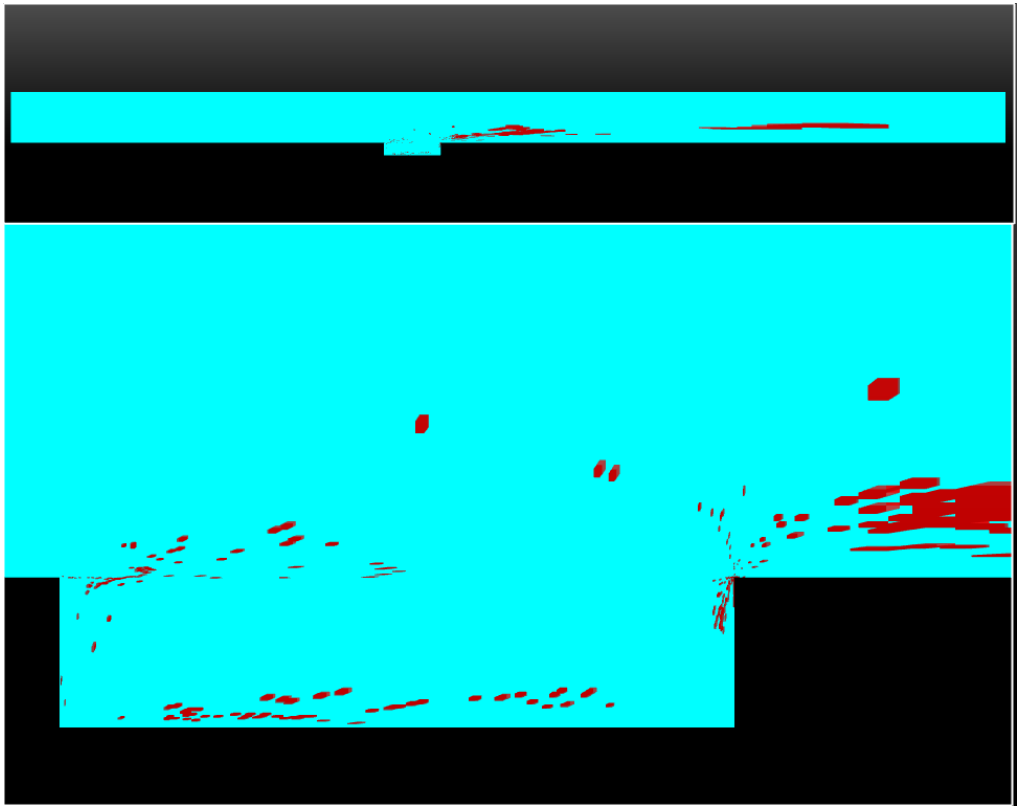
\+ ROMs are almost always *fast* ($p \ll n$).

# Example: Cavity-flow problem (collaborators: M Barone, S Arunajatesan)




- FOM: Compressible Navier–Stokes (AERO-F)
  - Nonlinear dynamical system
  - $\mathrm{Re} = 6.3 \times 10^6$
  - $M_\infty = 0.6$
  - DES turbulence model
  - 3 point BDF integrator
  - 1.2 million degrees of freedom
- ROM: GNAT [Carlberg et al., 2013]
  - 179 degrees of freedom

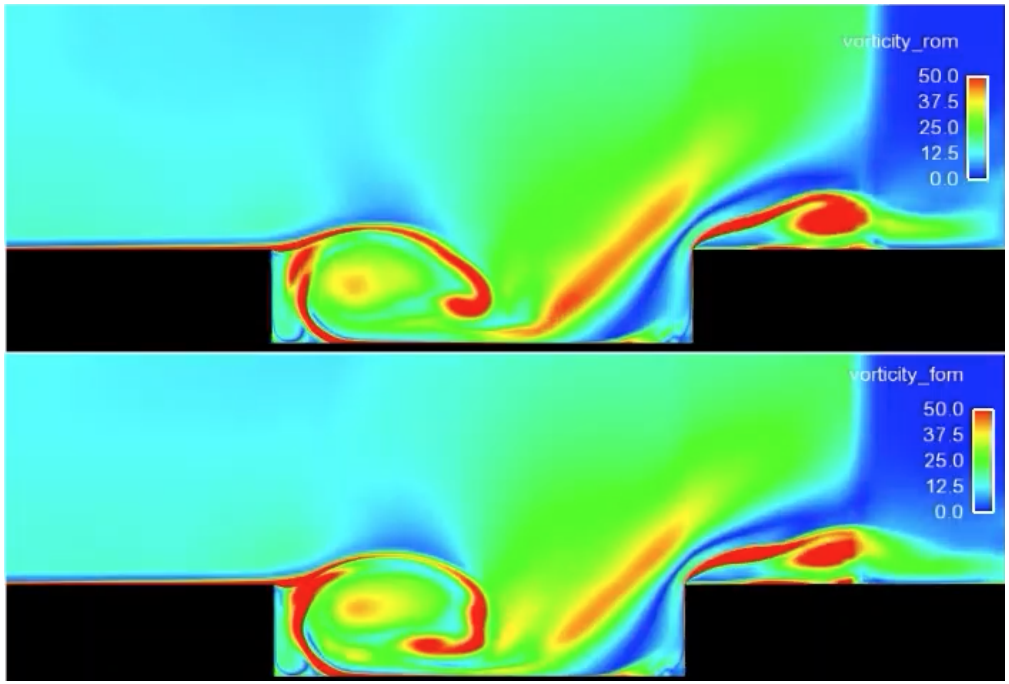
# Example: Cavity-flow problem (collaborators: M Barone, S Arunajatesan)



- sample mesh: 4% of original mesh

+ enables the ROM to run on far fewer cores

# When a ROM works




- FOM: 48 cpu x 5 hours
- ROM: 2 cpu x 31 min = 229X cpu savings
- However, ROMs are not guaranteed to be *accurate*.

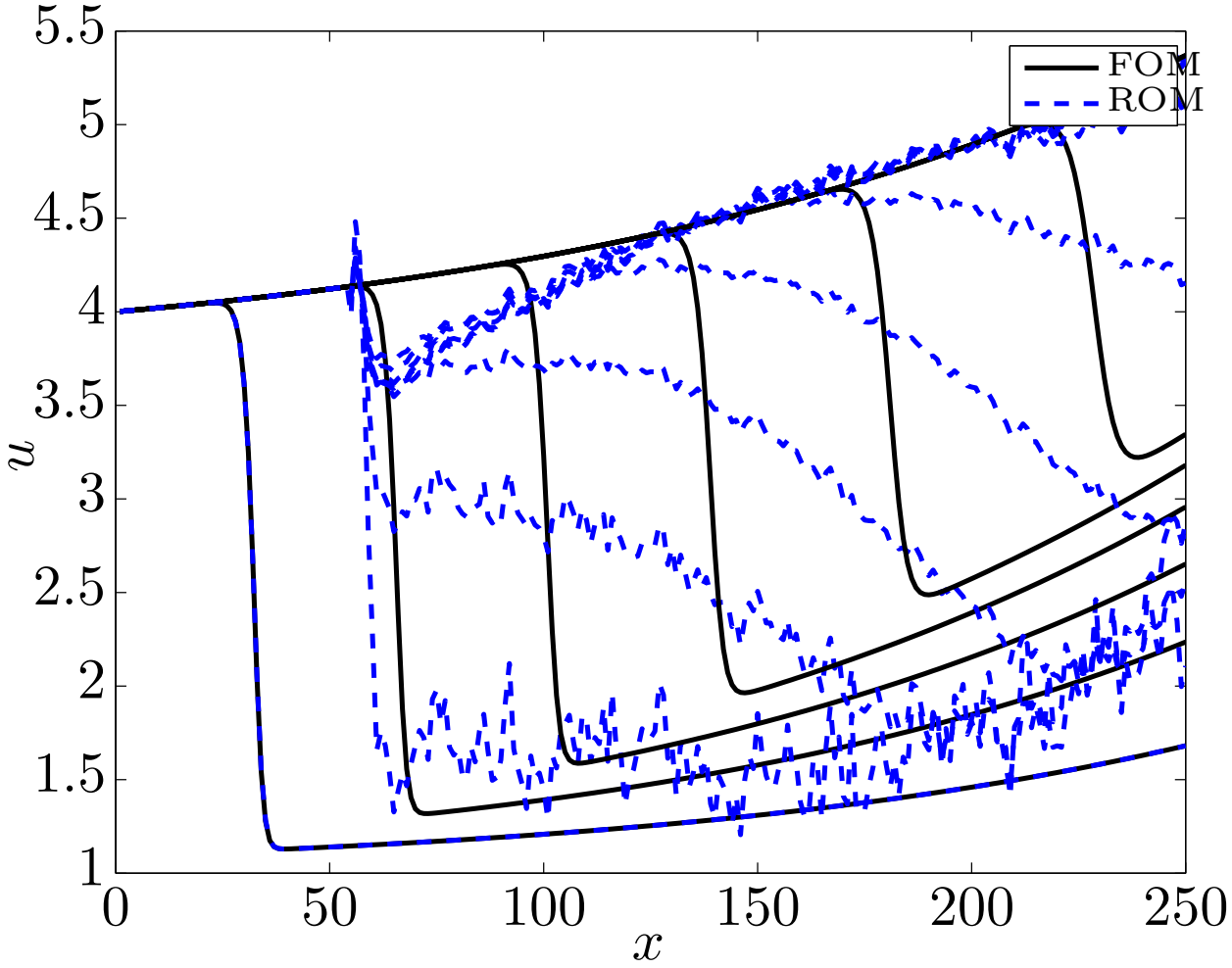
*ROM accuracy is limited by the information in* $\mathbf{V}$.

# Example: inviscid Burgers equation [Rewienski, 2003]

$$\frac{\partial u(x,\tau)}{\partial \tau} + \frac{1}{2}\frac{\partial\left(u^2(x,\tau)\right)}{\partial x} = 0.02e^{0.02x}$$
$$u(0,\tau) = 3,\ \forall \tau > 0$$
$$u(x,0) = 1,\ \forall x \in [0,\ 100]\,,$$

- Discretization: Godunov's scheme
- Simulate $\tau \in [0, 50]$
- FOM: 250 degrees of freedom
- ROM: 150 degrees of freedom
- **V** constructed via POD using snapshots in $\tau_{\text{train}} \in [0, 2.5]$

# ROM accuracy limited by relevance of training data




- ROM inaccurate when outside predictive domain of **V**

# Existing ROM adaptation methods

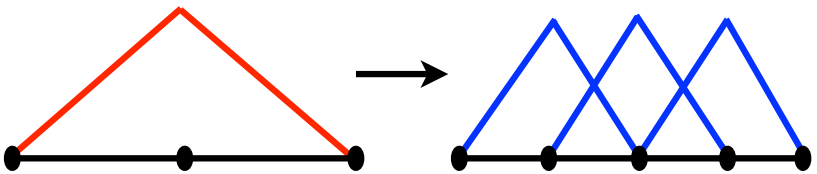
- *A priori* adaptation: unique ROM for separate regions of the
  - input space [Amsallem and Farhat, 2008, Amsallem et al., 2009, Eftang et al., 2010, Eftang et al., 2011, Haasdonk et al., 2011, Drohmann et al., 2011, Peherstorfer et al., 2014]
  - time domain [Drohmann et al., 2011, Dihlmann et al., ]
  - state space [Amsallem et al., 2012, Washabaugh et al., 2012, Peherstorfer et al., 2014].
  - \+ Reduces the dimension of the ROM
  - \- No mechanism to improve the ROM *a posteriori*
- *A posteriori* adaptation
  - Revert to the FOM, solve it, and add solution to the basis [Eldred et al., 2009, Arian et al., 2000, Ryckelynck, 2005]
  - Use ROM as a preconditioner for the FOM iterative solver [Carlberg and Farhat, 2009]
  - \+ Improves the ROM *a posteriori*
  - \- Incurs large-scale operations

**Goal**: *Cheap, a posteriori improvement of the ROM*

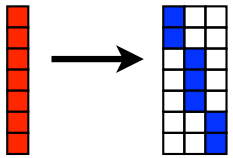
# Main idea

*ROM analog to mesh-adaptive h-refinement*

- 'Split' basis vectors

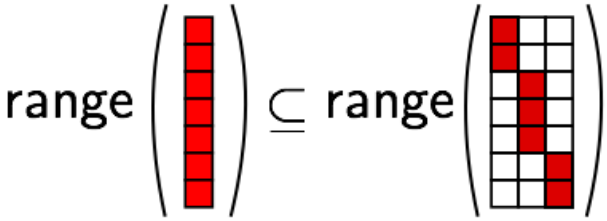


*finite element h-refinement*

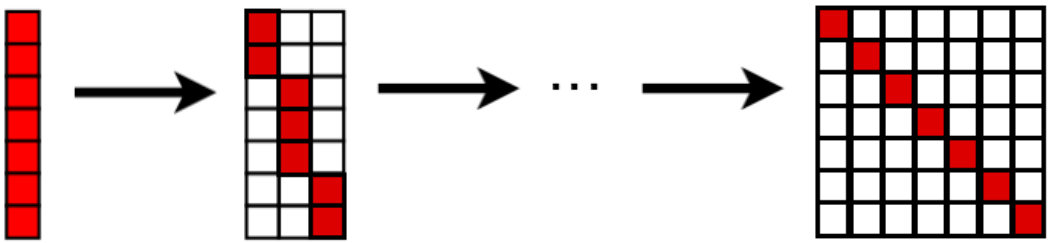


*ROM h-refinement*

- Generate hierarchical subspaces

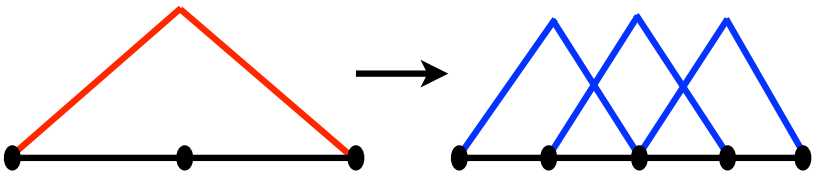


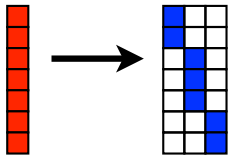

- ROM converges to the FOM



# *h*-refinement ingredients

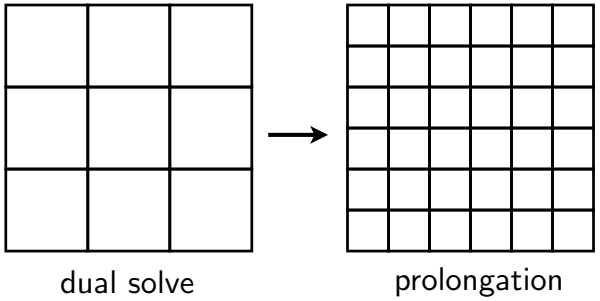
1. *Refinement*: split into basis vectors with different support

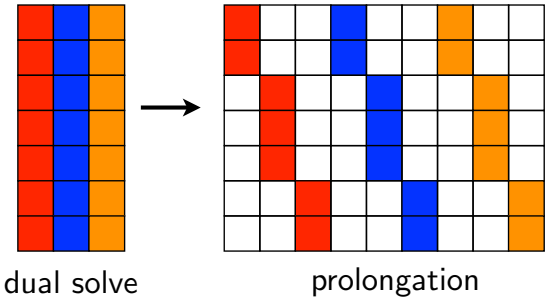




*finite element h-refinement* *ROM h-refinement*

2. *Error indicators*: a) dual solve (coarse), b) prolongation (fine)



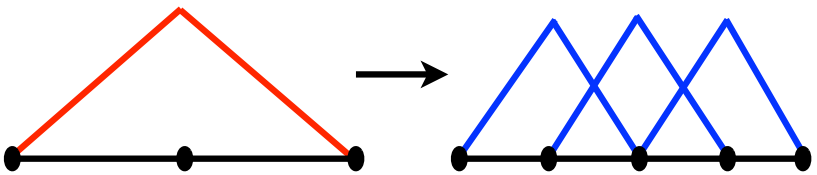




*finite element h-refinement* *ROM h-refinement*

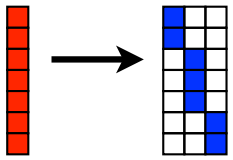
3. *Adaptive algorithm*: refine vectors with large error indicators

# Ingredient 1: Refinement

1. *Refinement*: split into basis vectors with different support

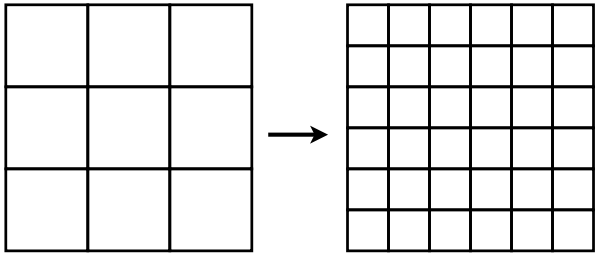


*finite element h-refinement*



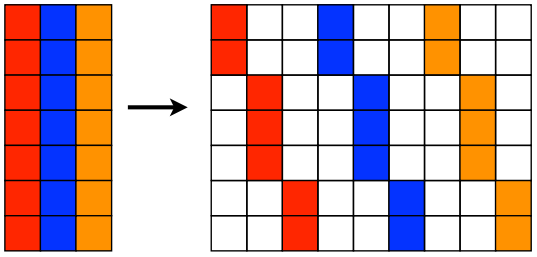
*ROM h-refinement*

2. *Error indicators*: a) dual solve (coarse), b) prolongation (fine)



dual solve prolongation

*finite element h-refinement*

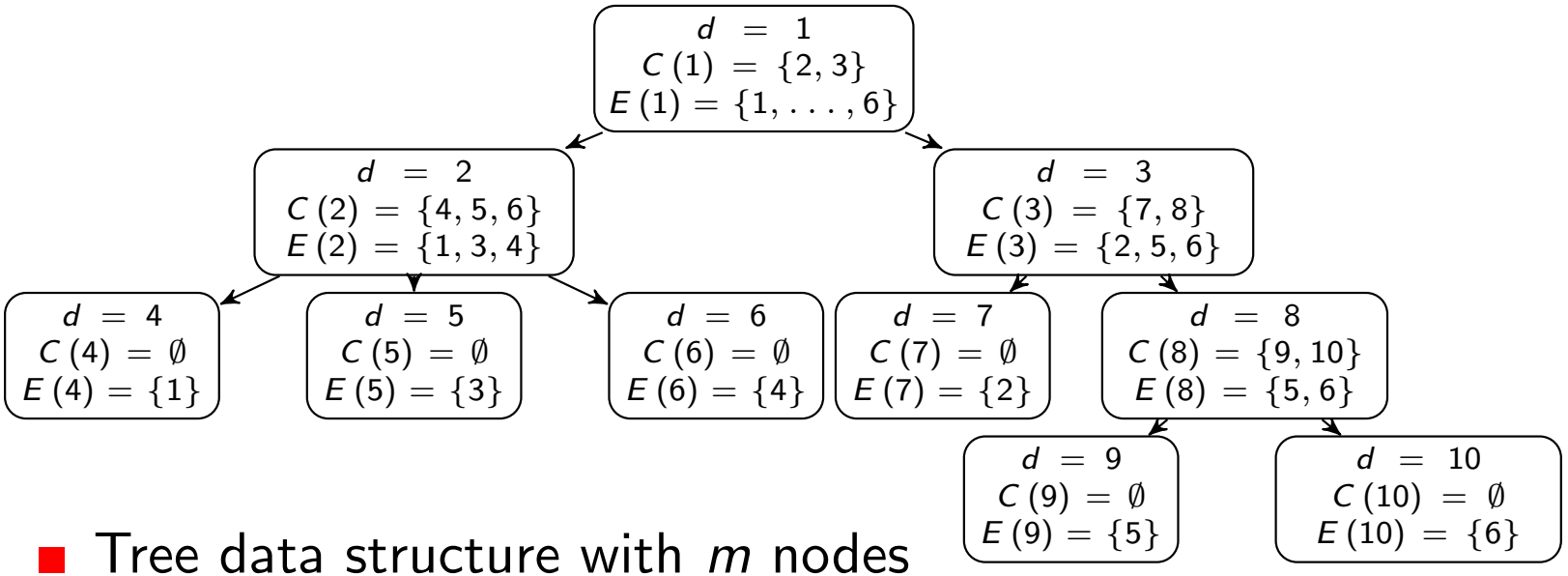


dual solve prolongation

*ROM h-refinement*

3. *Adaptive algorithm*: refine basis vecs with large error indicators

# Tree data structure

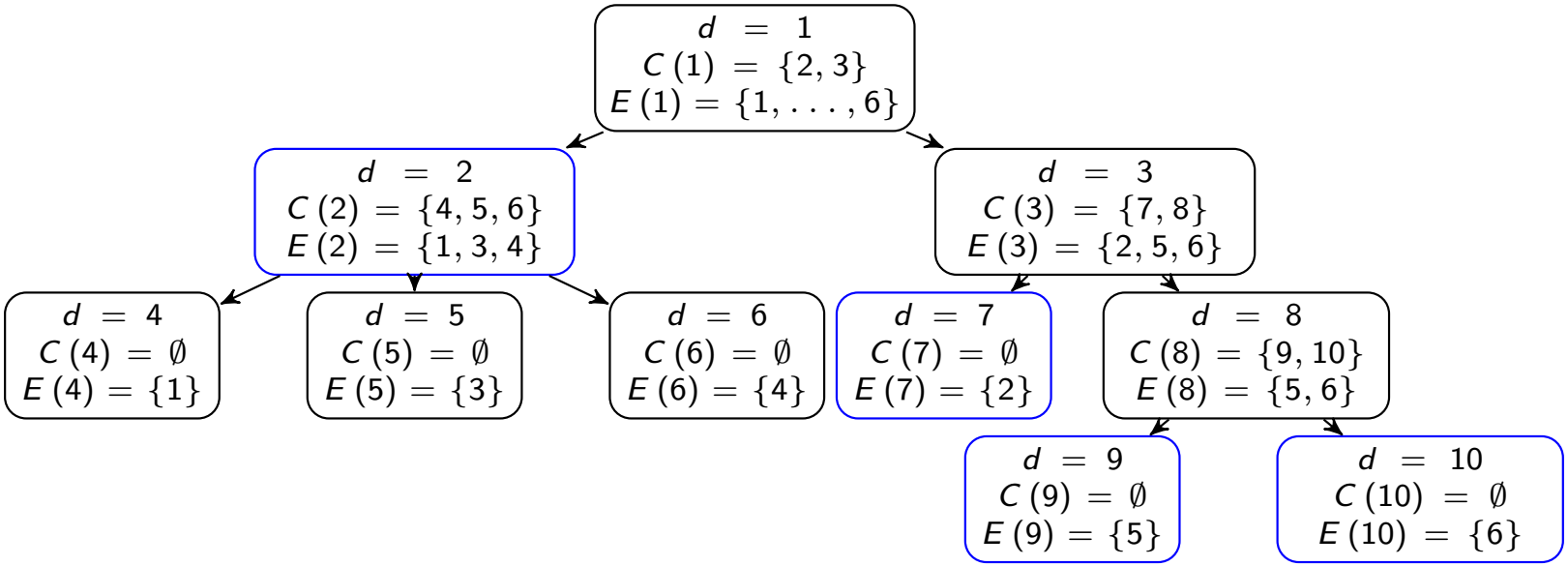



- Tree data structure with $m$ nodes
  - *child* function $C : \mathbb{N}(m) \to \mathcal{P}(\mathbb{N}(m))$
  - *element* function $E : \mathbb{N}(m) \to \mathcal{P}(\mathbb{N}(n))$
- Requirements
  1. Root node includes all elements $E(1)$
  2. Each element has a single leaf node
  3. Disjoint support of children $E(j) \cap E(k) = \emptyset, \ \forall j \neq k \in C(i)$
  4. $\cup_{j \in C(i)} E(j) = E(i)$

  \+ 1–2 ensure the ROM converges to the FOM

  \+ 4 ensures hierarchical refined subspaces

# Tree example: $n = 6$

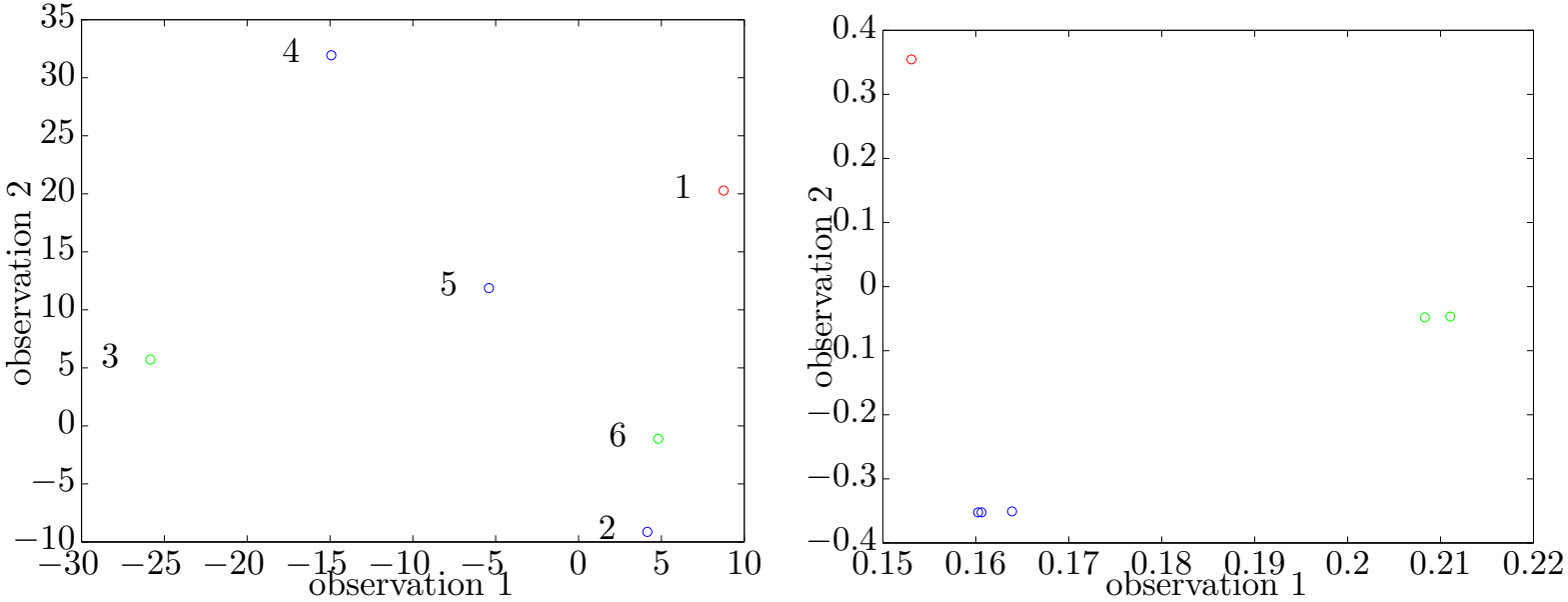



$$\mathbf{v}^{(0)} = \begin{bmatrix} v_1 \\ v_2 \\ v_3 \\ v_4 \\ v_5 \\ v_6 \end{bmatrix} \quad \rightarrow \quad \mathbf{V} = \begin{bmatrix} v_1 & 0 & 0 & 0 \\ 0 & v_2 & 0 & 0 \\ v_3 & 0 & 0 & 0 \\ v_4 & 0 & 0 & 0 \\ 0 & 0 & v_5 & 0 \\ 0 & 0 & 0 & v_6 \end{bmatrix}.$$

# Tree construction

*State variables that are strongly* **correlated** *or* **anticorrelated** *should reside in the same tree node.*

1. Normalize state-variable observation history
2. If first observation is negative, flip over origin
3. Recursively apply **k-means clustering**

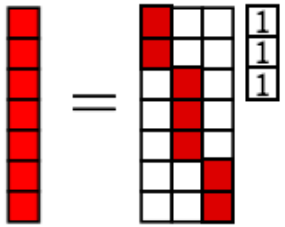



(a) before modification (b) after modification

Figure : State-variable observation history (variable index labeled).

# Refinement machinery

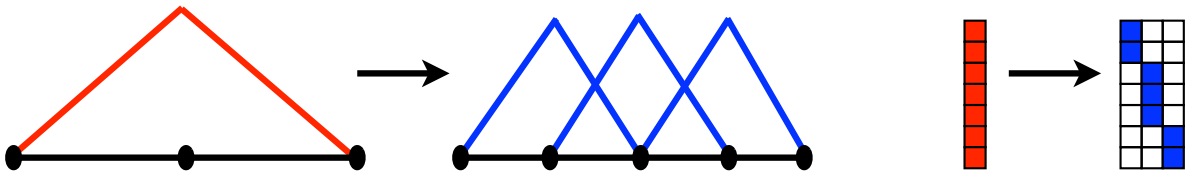
$$\mathbf{V}^H = \mathbf{V}^h \mathbf{I}_H^h$$




- coarse basis $\mathbf{V}^H \in \mathbb{R}^{n \times p}$ $\mathbf{V}^{(0)}$)
- fine basis $\mathbf{V}^h \in \mathbb{R}^{n \times q}$ with $q = \sum_{i=1}^{p} \text{card}\left(C\left(d_i\right)\right)$
- prolongation operator $\mathbf{I}_H^h \in \{0, 1\}^{q \times p}$
- prolongated generalized coordinates $\hat{\mathbf{x}}_H^h = \mathbf{I}_H^h \hat{\mathbf{x}}^H$
- restriction operator $\mathbf{I}_h^H = \left(\mathbf{I}_H^h\right)^+$

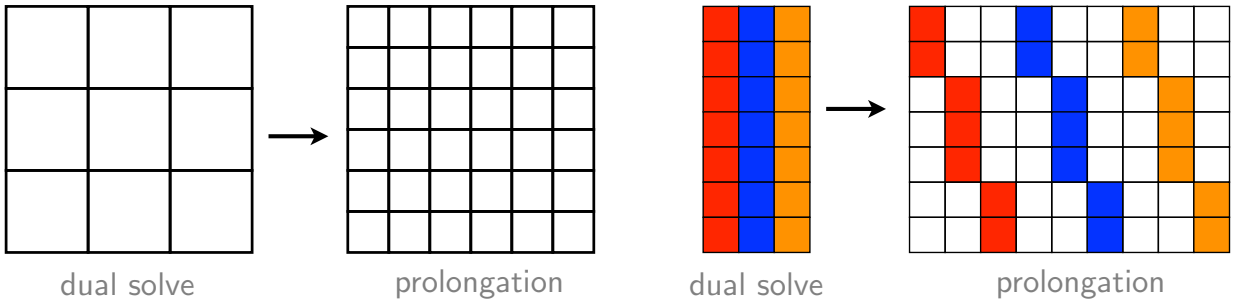
# Ingredient 2: Error indicators

- **Main idea**: ROM analog to mesh-adaptive *h*-refinement
  1. *Refinement*: split into basis vectors with different support



*finite element h-refinement* *ROM h-refinement*

  2. *Error indicators*: a) dual solve (coarse), b) prolongation (fine)




*finite element h-refinement* *ROM h-refinement*

  3. *Adaptive algorithm*: refine basis vecs with large error indicators

# Dual-weighted residual error indicators

- **Goal-oriented**: reduce the error in output $z(\mathbf{x})$
- Analogous to duality-based error control for
  - differential equations [Estep, 1995, Pierce and Giles, 2000]
  - finite elements [Babuška and Miller, 1984, Becker and Rannacher, 1996, Rannacher, 1999, Bangerth and Rannacher, 1999, Becker and Rannacher, 2001, Bangerth and Rannacher, 2003],
  - finite volumes [Venditti and Darmofal, 2000, Venditti and Darmofal, 2002, Park, 2004, Nemec and Aftosmis, 2007]
  - discontinuous Galerkin methods [Lu, 2005, Fidkowski, 2007]

# Dual-weighted residual error indicators

- Approximate fine output:

$$z(\mathbf{V}^h\hat{\mathbf{x}}^h) \approx z(\mathbf{V}^H\hat{\mathbf{x}}^H) + \frac{\partial z}{\partial \mathbf{x}}(\mathbf{V}^H\hat{\mathbf{x}}^H)\mathbf{V}^h(\hat{\mathbf{x}}^h - \mathbf{I}_H^h\hat{\mathbf{x}}^H) \qquad (1)$$

- Approximate the fine residual:

$$0 = (\mathbf{V}^h)^T\mathbf{r}(\mathbf{V}^h\hat{\mathbf{x}}^h) \approx (\mathbf{V}^h)^T\mathbf{r}(\mathbf{V}^H\hat{\mathbf{x}}^H) + (\mathbf{V}^h)^T\frac{\partial \mathbf{r}}{\partial \mathbf{x}}(\mathbf{V}^H\hat{\mathbf{x}}^H)\mathbf{V}^h(\hat{\mathbf{x}}^h - \mathbf{I}_H^h\hat{\mathbf{x}}^H)$$

- Solve for the error:

$$(\hat{\mathbf{x}}^h - \mathbf{I}_H^h\hat{\mathbf{x}}^H) \approx -[(\mathbf{V}^h)^T\frac{\partial \mathbf{r}}{\partial \mathbf{x}}(\mathbf{V}^H\hat{\mathbf{x}}^H)\mathbf{V}^h]^{-1}(\mathbf{V}^h)^T\mathbf{r}(\mathbf{V}^H\hat{\mathbf{x}}^H) \qquad (2)$$

- Substitute (2) in (1):

$$\boxed{z(\mathbf{V}^h\hat{\mathbf{x}}^h) - z(\mathbf{V}^H\hat{\mathbf{x}}^H) \approx -(\hat{\mathbf{y}}^h)^T(\mathbf{V}^h)^T\mathbf{r}(\mathbf{V}^H\hat{\mathbf{x}}^H)},$$

with the fine adjoint solution $\hat{\mathbf{y}}^h \in \mathbb{R}^q$ satisfying

$$\boxed{(\mathbf{V}^{h\,T}\frac{\partial \mathbf{r}^k}{\partial \mathbf{x}}(\mathbf{V}^H\hat{\mathbf{x}}^H)^T\mathbf{V}^h\hat{\mathbf{y}}^h = \mathbf{V}^{h\,T}\frac{\partial z}{\partial \mathbf{x}}(\mathbf{V}^H\hat{\mathbf{x}}^H)^T}.$$

# Dual-weighted residual error indicators

$$\boxed{z(\mathbf{V}^h\hat{\mathbf{x}}^h) - z(\mathbf{V}^H\hat{\mathbf{x}}^H) \approx -(\hat{\mathbf{y}}^h)^T(\mathbf{V}^h)^T\mathbf{r}(\mathbf{V}^H\hat{\mathbf{x}}^H)} \tag{3}$$

$$\boxed{(\mathbf{V}^h)^T\frac{\partial \mathbf{r}^k}{\partial \mathbf{x}}(\mathbf{V}^H\hat{\mathbf{x}}^H)^T\mathbf{V}^h\hat{\mathbf{y}}^h = (\mathbf{V}^h)^T\frac{\partial z}{\partial \mathbf{x}}(\mathbf{V}^H\hat{\mathbf{x}}^H)^T} \tag{4}$$

- We want to avoid fine solve (4), so approximate $\hat{\mathbf{y}}^h$ as

$$\hat{\mathbf{y}}^h_H = \mathbf{I}^h_H\hat{\mathbf{y}}^H,$$

  where $\hat{\mathbf{y}}^H$ is the coarse adjoint solution to

$$(\mathbf{V}^H)^T\frac{\partial \mathbf{r}^k}{\partial \mathbf{x}}(\mathbf{V}^H\hat{\mathbf{x}}^H)^T\mathbf{V}^H\hat{\mathbf{y}}^H = (\mathbf{V}^H)^T\frac{\partial z}{\partial \mathbf{x}}(\mathbf{V}^H\hat{\mathbf{x}}^H)^T.$$

- Substituting $\hat{\mathbf{y}}^h_H$ for $\hat{\mathbf{y}}^h$ in (3) yields cheaply computable

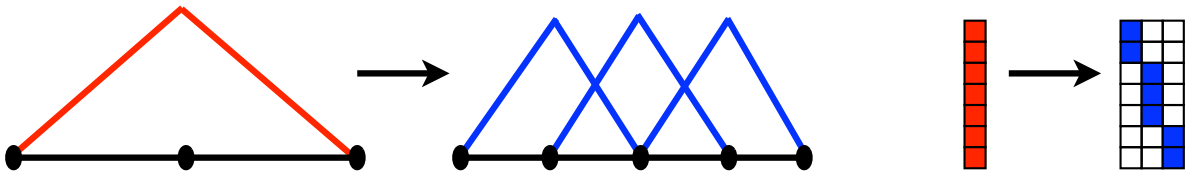
$$z(\mathbf{V}^h\hat{\mathbf{x}}^h) - z(\mathbf{V}^H\hat{\mathbf{x}}^H) \approx -(\hat{\mathbf{y}}^h_H)^T(\mathbf{V}^h)^T\mathbf{r}(\mathbf{V}^H\hat{\mathbf{x}}^H).$$

- The RHS can be bounded by cheaply computable error indicators

$$|(\hat{\mathbf{y}}^h_H)^T(\mathbf{V}^h)^T\mathbf{r}(\mathbf{V}^H\hat{\mathbf{x}}^H)| \leq \sum_{i=1}^{q}\boldsymbol{\Delta}^h_i, \quad \boxed{\boldsymbol{\Delta}^h_i = |\left[\hat{y}^h_H\right]_i\left(\mathbf{v}^h_i\right)^T\mathbf{r}\left(\mathbf{V}^H\hat{\mathbf{x}}^H\right)|.}$$

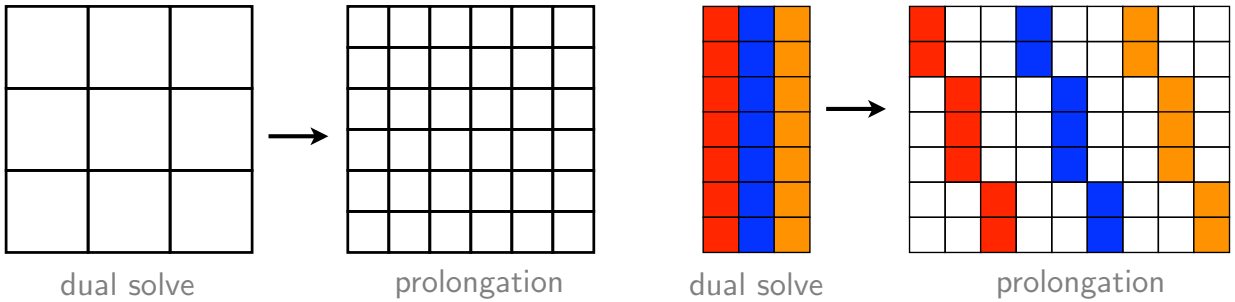
# Ingredient 3: Adaptive algorithm

- **Main idea**: ROM analog to mesh-adaptive *h*-refinement
  1. *Refinement*: split into basis vectors with different support



*finite element h-refinement* *ROM h-refinement*

  2. *Error indicators*: a) dual solve (coarse), b) prolongation (fine)




*finite element h-refinement* *ROM h-refinement*

  3. *Adaptive algorithm*: refine basis vecs with large error indicators

# Adaptive algorithm

**Algorithm 1** Outer loop

**Input:** timestep $k$, basis $\mathbf{V}$
**Output:** updated basis $\mathbf{V}$, generalized state $\hat{\mathbf{x}}^k$

1: Compute ROM solution $\hat{\mathbf{x}}^k$ satisfying $\mathbf{V}^T\mathbf{r}^k(\mathbf{V}\hat{\mathbf{x}}^k;\boldsymbol{\mu}) = 0$.
2: **if** estimate of output error $\delta_{\mathbf{z}}$ is 'too large' **then**
3: Refine basis via Algorithm 2: $\mathbf{V} \leftarrow \text{Refine}\left(\mathbf{V}, \hat{\mathbf{x}}^k\right)$.
4: Return to Step 1.
5: **end if**
6: **if** $\text{mod}\left(k, n_{\text{reset}}\right) = 0$ **then**
7: Reset basis $\mathbf{V} \leftarrow \mathbf{V}^{(0)}$.
8: **end if**

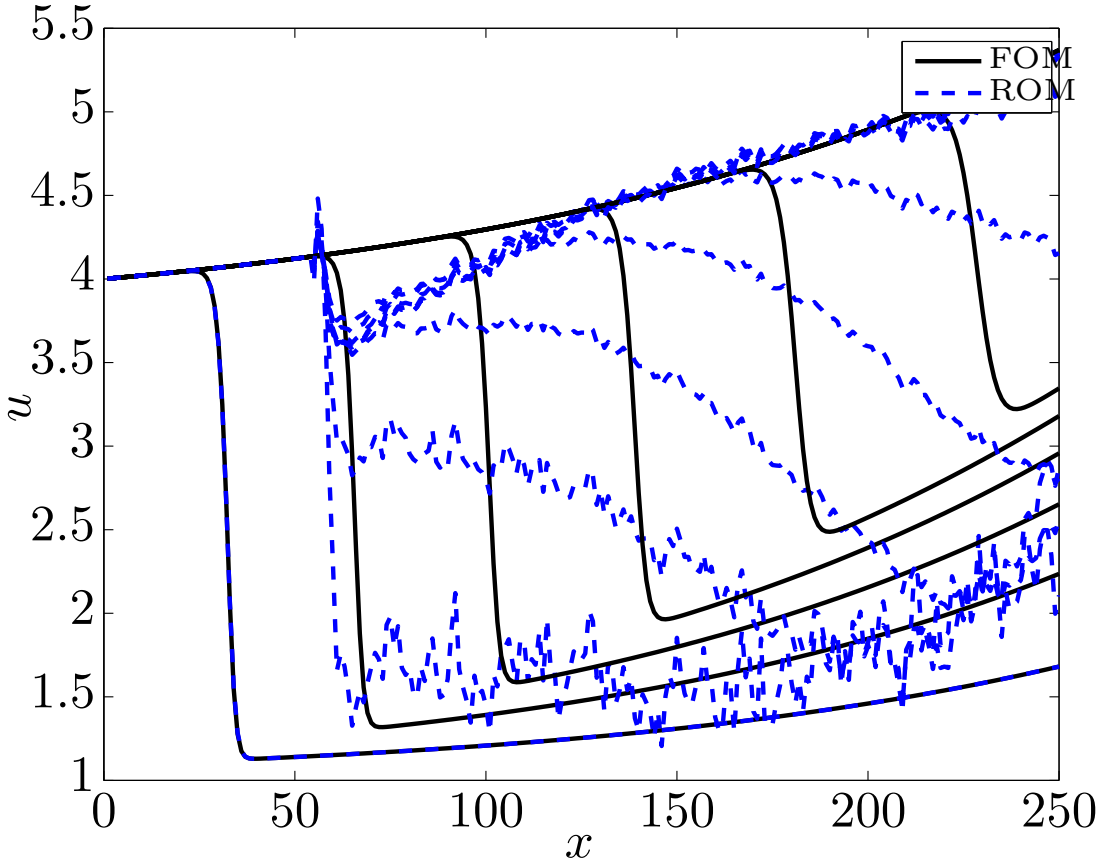
# Adaptive algorithm

**Algorithm 2** Refine

**Input:** initial basis $\mathbf{V}$, reduced solution $\hat{\mathbf{x}}$
**Output:** refined basis $\mathbf{V}$

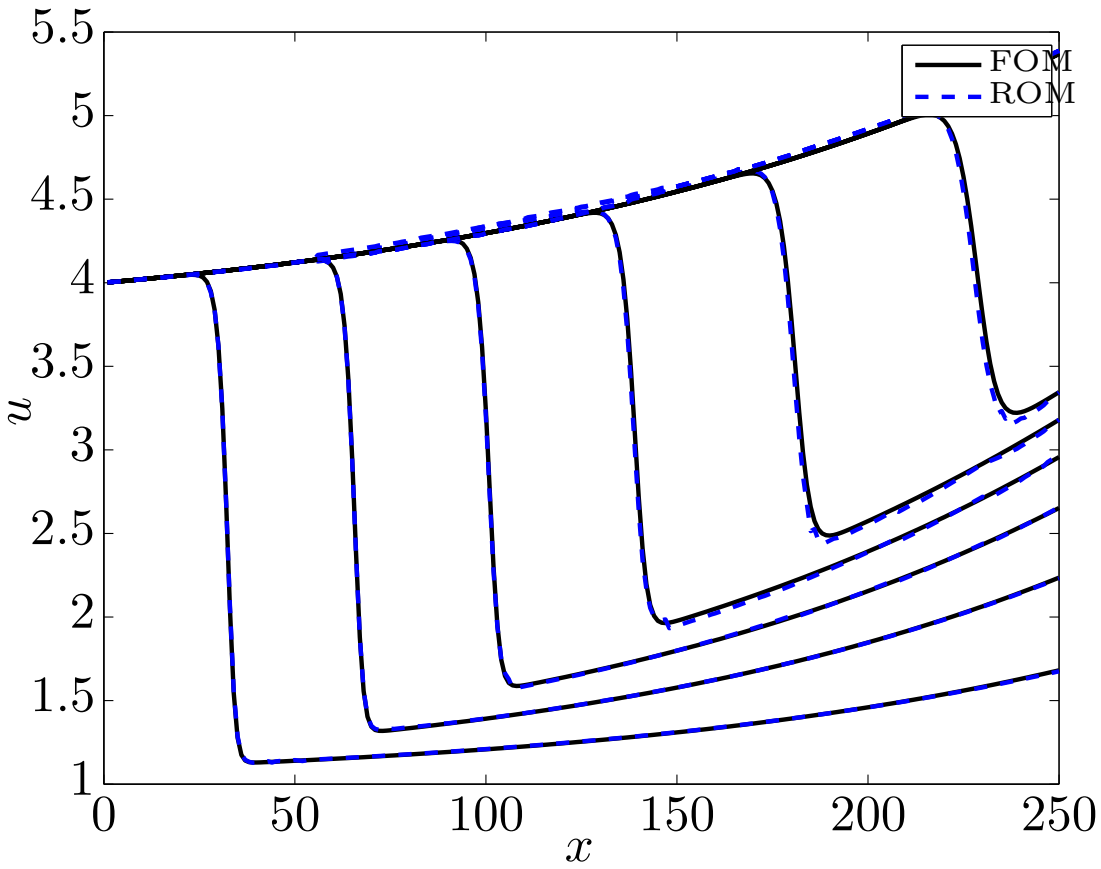
1: Compute prolongation operator $\mathbf{I}_H^h$ and fine basis $\mathbf{V}^h$
2: Solve: Compute coarse adjoint solution $\hat{\mathbf{y}}^H$ and prolongation $\hat{\mathbf{y}}_H^h$
3: Estimate: Compute fine error indicators $\boldsymbol{\Delta}_i^h$, $i = 1, \ldots, q$
4: Mark: Identify basis vectors to refine $I$
5: **for** $i \in I$ **do**
6: Refine: Split $\mathbf{v}_i$ into $\mathrm{card}\left(C\left(d_i\right)\right)$ vectors
7: **end for**
8: Compute QR factorization with column pivoting $\mathbf{V} = \mathbf{QR}$, $\mathbf{R}\bar{\mathbf{\Pi}} = \bar{\mathbf{Q}}\bar{\mathbf{R}}$
9: Ensure full-rank matrix $\mathbf{V} \leftarrow \mathbf{V}\left[\bar{\boldsymbol{\pi}}_1 \ \cdots \ \bar{\boldsymbol{\pi}}_r\right]$

# Previous example




- Generated by $\mathbf{V} \in \mathbb{R}^{250\times150}$ using $\tau_{\text{train}} \in [0, 2.5]$
- Now try *h*-adaptivity with $\mathbf{V}^{(0)} \in \mathbb{R}^{250\times10}$, $\tau_{\text{train}} \in [0, 2.5]$.

# Previous example with *h*-adaptivity (FOM tolerance $\epsilon = 0.05$)




- $\dim \mathbf{V}^{(0)} = 10$
- $\text{mean}(\dim \mathbf{V}) = 44.3$

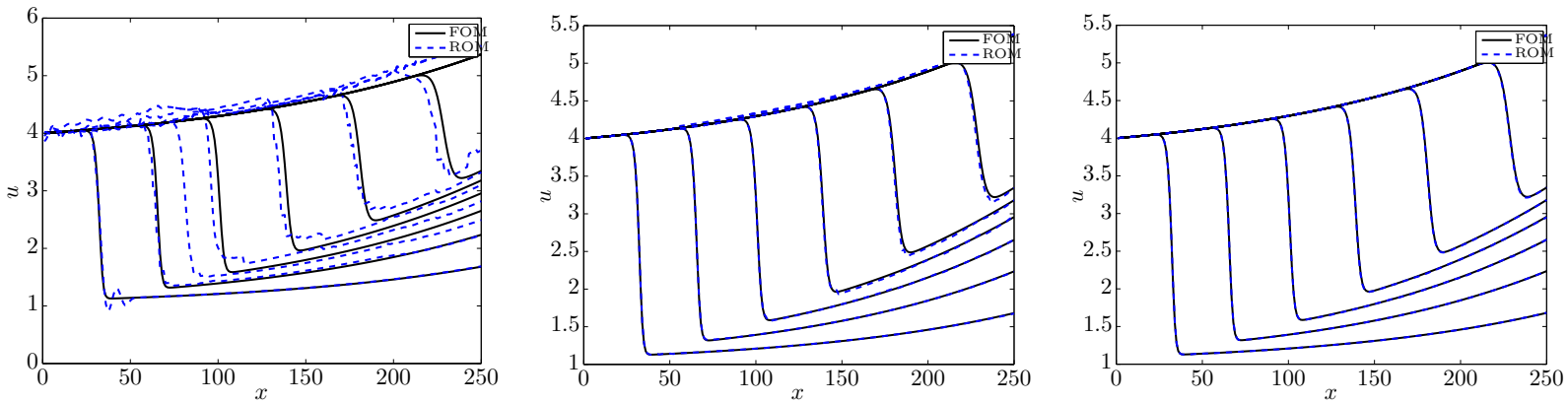
*h-adaptation allows the ROM to capture phenomena not present in the training data!*

# Inviscid Burgers equation: results

| | typical ROM | | | *h*-adaptive ROM | | | | |
|---|---|---|---|---|---|---|---|---|
| dim $\mathbf{V}^{(0)}$ | 10 | 45 | 150 | 5 | 10 | 20 | 10 | 10 |
| basis-reset frequency | | | | 50 | 50 | 50 | 100 | 25 |
| mean(dim $\mathbf{V}$) | 10 | 45 | 150 | 41.4 | 44.3 | 58 | 73 | 37 |
| Avg splits | | | | 0.20 | 0.19 | 0.14 | 0.13 | 0.28 |
| error (%) | 45.8 | 43.9 | 8.5 | 0.3 | 0.5 | 0.2 | 0.2 | 0.3 |
| online time (s) | 1.4 | 2.14 | 5.77 | 5.53 | 4.63 | 7.27 | 6.90 | 7.46 |

+ Low errors achievable only with *h*-adaptivity
- *Smaller initial basis*: smaller basis dimension, but more splits
- *More frequent basis resetting*: smaller basis dimension, but more splits

# $h$-adaptivity enables uncertainty/error control




(a) tolerance $\epsilon = 0.35$ (b) tolerance $\epsilon = 0.05$ (c) tolerance $\epsilon = 0.01$

| | $\epsilon = 0.35$ | $\epsilon = 0.05$ | $\epsilon = 0.01$ |
|---|---|---|---|
| average basis dimension per Newton iteration | 33.6 | 44.2507 | 53.9 |
| relative error (%) | 12.2 | 0.51 | 0.078 |
| online time (seconds) | 4.61 | 4.63 | 7.64 |

# Conclusions

- Adaptive *h*-refinement via splitting
  - \+ Incrementally improves ROM
  - \+ Does not require large-scale operations
  - \+ Enables uncertainty/error control
  - \+ Extends utility of ROMs to hyperbolic PDEs
- Future work
  - Production implementation
  - Adaptive *coarsening*
  - Adaptive *p*-refinement: add basis vectors from a library

# Error modeling

$$\bar{\mathbf{z}} = \mathbf{z}_{\text{red}}(\boldsymbol{\mu}^\star) + \boxed{\delta_{\mathbf{z}}(\boldsymbol{\mu}^\star)} + \varepsilon$$

*"An interesting future research direction is the inclusion of estimates of reduced model error as an additional source of uncertainty in the Bayesian formulation."*

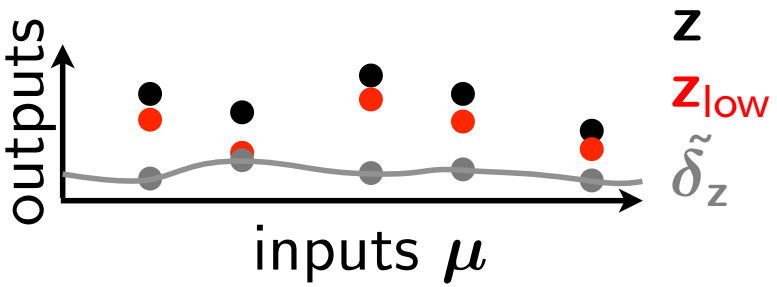
*[Galbally et al., 2009]*

**Our goals**:

1. construct a statistical model of this error
2. control this error

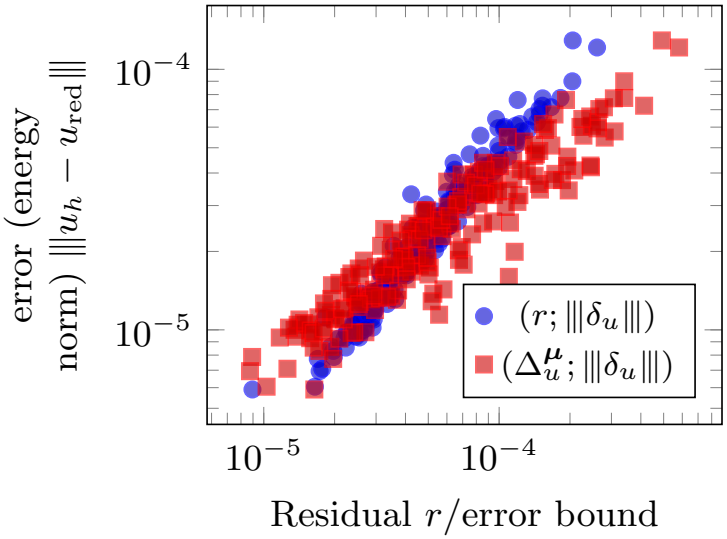
# Strategies for ROM error quantification

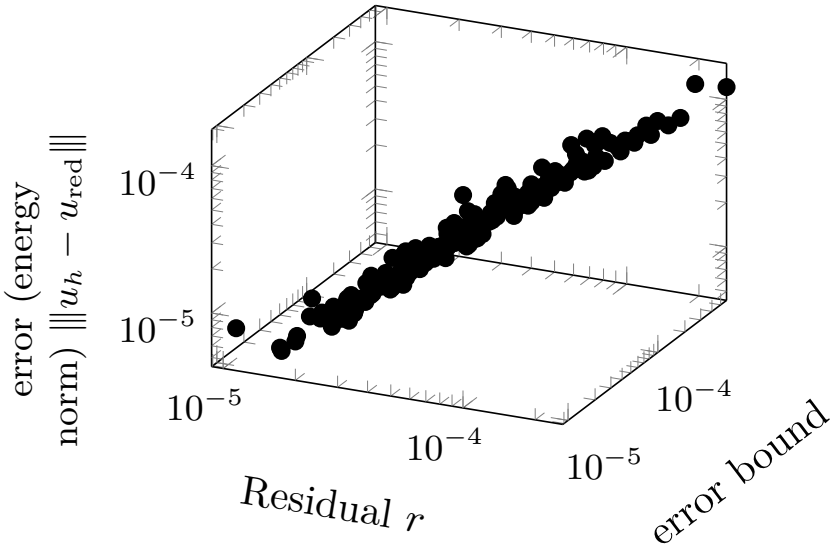
1. Rigorous error bounds
   - \+ independent of input-space dimension
   - \- not amenable to statistics
   - \- high effectivity (overestimation)
   - \- improving effectivity incurs high costs [Huynh et al., 2010, Wirtz et al., 2012] or intrusive reformulation of discretization [Yano et al., 2012]
2. Multifidelity correction [Eldred et al., 2004]




- model low-fidelity error as a function of inputs
- 'correct' low-fidelity outputs with model
- \+ amenable to statistics
- \- curse of dimensionality
- \- ROM errors highly oscillatory in the input space [Ng and Eldred, 2012]

# Our key observation

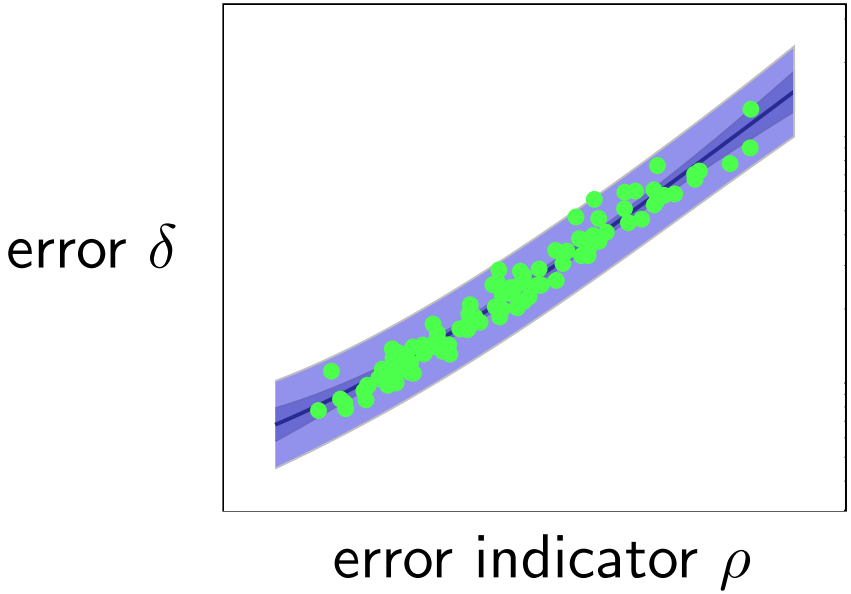



- ROMs generate **error indicators** that correlate with the error
- **Main idea**: construct a stochastic process that maps error indicators (not inputs $\boldsymbol{\mu}$!) to a distribution over the true error
  - \+ independent of input-space dimension
  - \+ quantifies ROM **epistemic uncertainty**

*Reduced-order model error surrogates*

# ROMES




- Construct a stochastic process of the ROM error $\widetilde{\delta}(\boldsymbol{\rho})$
- Select a *small number* of error indicators $\boldsymbol{\rho} = \boldsymbol{\rho}(\boldsymbol{\mu})$ that are
  1. cheaply computable online, and
  2. lead to low variance of the stochastic process.
- First attempt: Gaussian process (GP) such that random variables $(\widetilde{\delta}(\boldsymbol{\rho}_1), \widetilde{\delta}(\boldsymbol{\rho}_2), \ldots)$ have joint Gaussian distribution

# Gaussian process [Rasmussen and Williams, 2006]

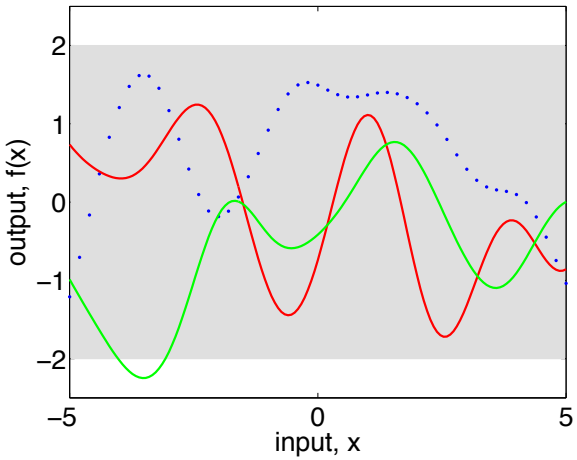
**Definition (Gaussian process)**

A Gaussian process is a collection of random variables, any finite number of which have a joint Gaussian distribution.

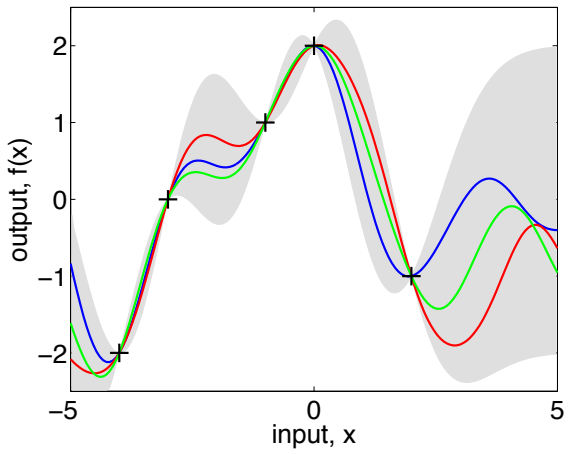
$$\widetilde{\delta}(\boldsymbol{\rho}) \sim \mathcal{GP}(m(\boldsymbol{\rho}), k(\boldsymbol{\rho}, \boldsymbol{\rho}'))$$

- mean function $m(\boldsymbol{\rho})$; covariance function $k(\boldsymbol{\rho}, \boldsymbol{\rho}')$
- given a training set $\{(\delta_i, \boldsymbol{\rho}_i)\}$, can infer $m(\boldsymbol{\rho})$ and $k(\boldsymbol{\rho}, \boldsymbol{\rho}')$
- consider two types of GPs
  1. kernel regression [Rasmussen and Williams, 2006]
  2. relevance vector machine (RVM) [Tipping, 2001]

# GP #1: Kernel regression [Rasmussen and Williams, 2006]




(d) prior




(e) posterior

- **prior**: $\widetilde{\delta}(\underline{\boldsymbol{\rho}}) \sim \mathcal{N}(0, K\left(\underline{\boldsymbol{\rho}}, \underline{\boldsymbol{\rho}}\right) + \sigma^2 \mathbf{I})$
  - $k(\boldsymbol{\rho}_i, \boldsymbol{\rho}_j) = \exp \frac{\|\boldsymbol{\rho}_i - \boldsymbol{\rho}_j\|^2}{r^2}$ is a positive definite kernel
  - $\underline{\boldsymbol{\rho}} := \left[\underline{\boldsymbol{\rho}}_{\text{train}} \ \underline{\boldsymbol{\rho}}_{\text{predict}}\right]^T$
- **posterior**: $\widetilde{\delta}(\boldsymbol{\rho}_{\text{predict}}) \sim \mathcal{N}(m(\boldsymbol{\rho}_{\text{predict}}), \mathrm{cov}(\boldsymbol{\rho}_{\text{predict}}))$
- infer hyperparameters $\sigma^2$ and $r^2$

# GP #2: Relevance vector machine [Tipping, 2001]

$$\widetilde{\delta}(\boldsymbol{\rho}) = \sum_{m=1}^{M} w_m \phi_m(\boldsymbol{\rho}) + \varepsilon$$

- fixed basis functions $\phi_m$ (e.g., polynomials)
- stochastic coefficients $w_m$
- noise $\varepsilon \sim \mathcal{N}(0, \sigma^2 \mathbf{I})$
- **prior**: $w \sim \mathcal{N}(0, \mathrm{diag}(\alpha_i))$
- **posterior**: $w \sim \mathcal{N}(\mathbf{m}, \boldsymbol{\Sigma})$ leads to posterior dist. of $\widetilde{\delta}(\boldsymbol{\rho})$
- infer hyperparameters $\sigma^2$ and $\alpha_i$

# ROMES Algorithm

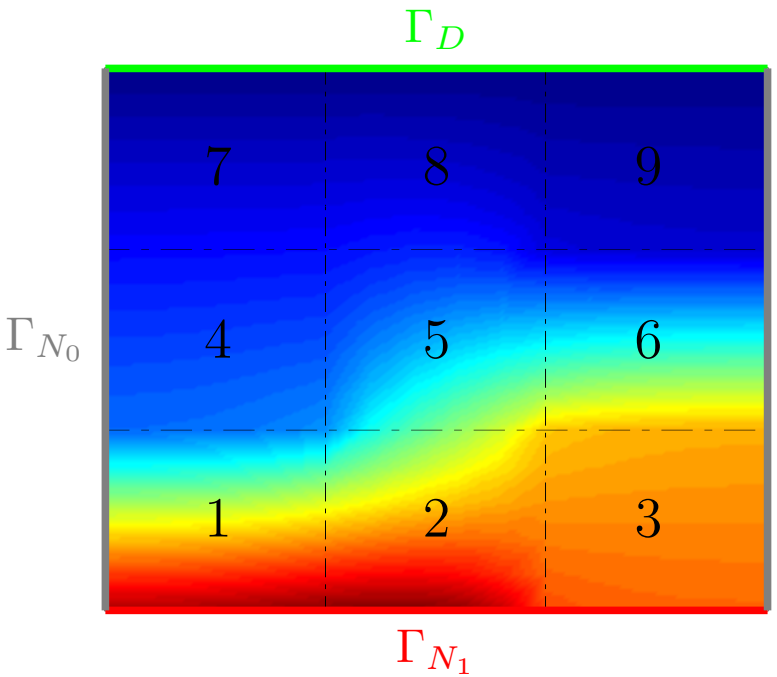
Offline

1. Populate ROMES database $\{(\delta(\boldsymbol{\mu}), \bar{\boldsymbol{\rho}}(\boldsymbol{\mu})) \mid \boldsymbol{\mu} \in \mathcal{D}_{\text{train}}\}$, where $\bar{\boldsymbol{\rho}}$ denotes candidate indicators.
2. Identify a few error indicators $\boldsymbol{\rho} \subset \bar{\boldsymbol{\rho}}$ that lead to a low-variance GP.
3. Construct the Gaussian process $\widetilde{\delta}(\boldsymbol{\rho}) \sim \mathcal{GP}(m(\boldsymbol{\rho}), k(\boldsymbol{\rho}, \boldsymbol{\rho}'))$ by Bayesian inference.

Online (for any $\boldsymbol{\mu}^\star \in \mathcal{D}$)

1. compute the ROM solution
2. compute error indicators $\boldsymbol{\rho}(\boldsymbol{\mu}^\star)$
3. obtain $\widetilde{\delta}(\boldsymbol{\rho}(\boldsymbol{\mu}^\star)) \sim \mathcal{N}(m(\boldsymbol{\rho}(\boldsymbol{\mu}^\star))), k(\boldsymbol{\rho}(\boldsymbol{\mu}^\star), \boldsymbol{\rho}(\boldsymbol{\mu}^\star))$
4. correct the ROM solution

# Thermal block (Parametrically coercive and compliant, affine, linear, elliptic)

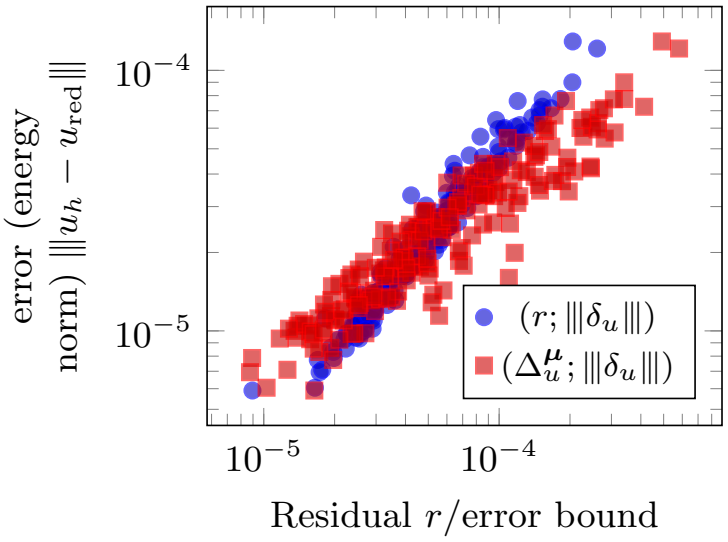


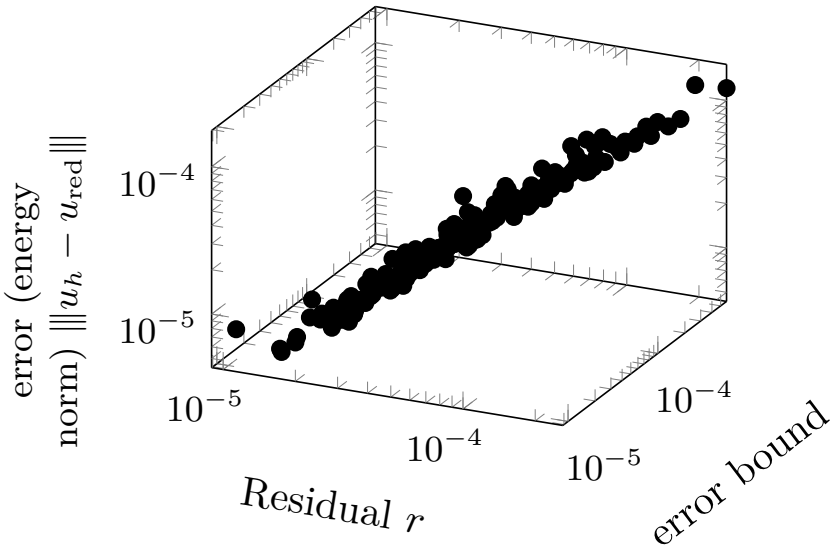

$$\triangle c(x;\boldsymbol{\mu})u(x;\boldsymbol{\mu}) = 0 \text{ in } \Omega \qquad \mathbf{x}(\boldsymbol{\mu}) = 0 \text{ on } \Gamma_D$$

$$\nabla c(\boldsymbol{\mu})\mathbf{x}(\boldsymbol{\mu}) \cdot n = 0 \text{ on } \Gamma_{N_0} \qquad \nabla c(\boldsymbol{\mu})\mathbf{x}(\boldsymbol{\mu}) \cdot n = 1 \text{ on } \Gamma_{N_1}$$

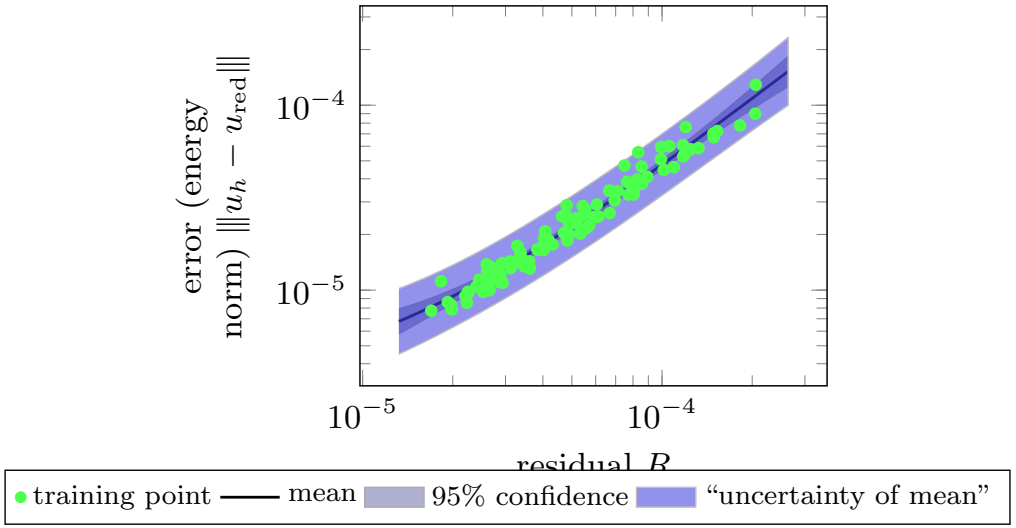
- Inputs $\boldsymbol{\mu} \in [0.1, 10]^9$ define diffusivity $c$ in subdomains
- Output $z(\boldsymbol{\mu}) = \int_{\Gamma_{N_1}} \mathbf{x}(\boldsymbol{\mu})\mathrm{d}x$ is compliant
- ROM constructed via RB–Greedy [Patera and Rozza, 2006]
- Consider two errors: 1) energy norm of state-space error $|||\mathbf{x}(\boldsymbol{\mu}) - \hat{\mathbf{x}}(\boldsymbol{\mu})|||$, 2) output bias $\mathbf{z}(\boldsymbol{\mu}) - \mathbf{z}_{\text{red}}(\boldsymbol{\mu})$

# Error #1: energy norm $|||\mathbf{x}(\boldsymbol{\mu}) - \hat{\mathbf{x}}(\boldsymbol{\mu})|||$



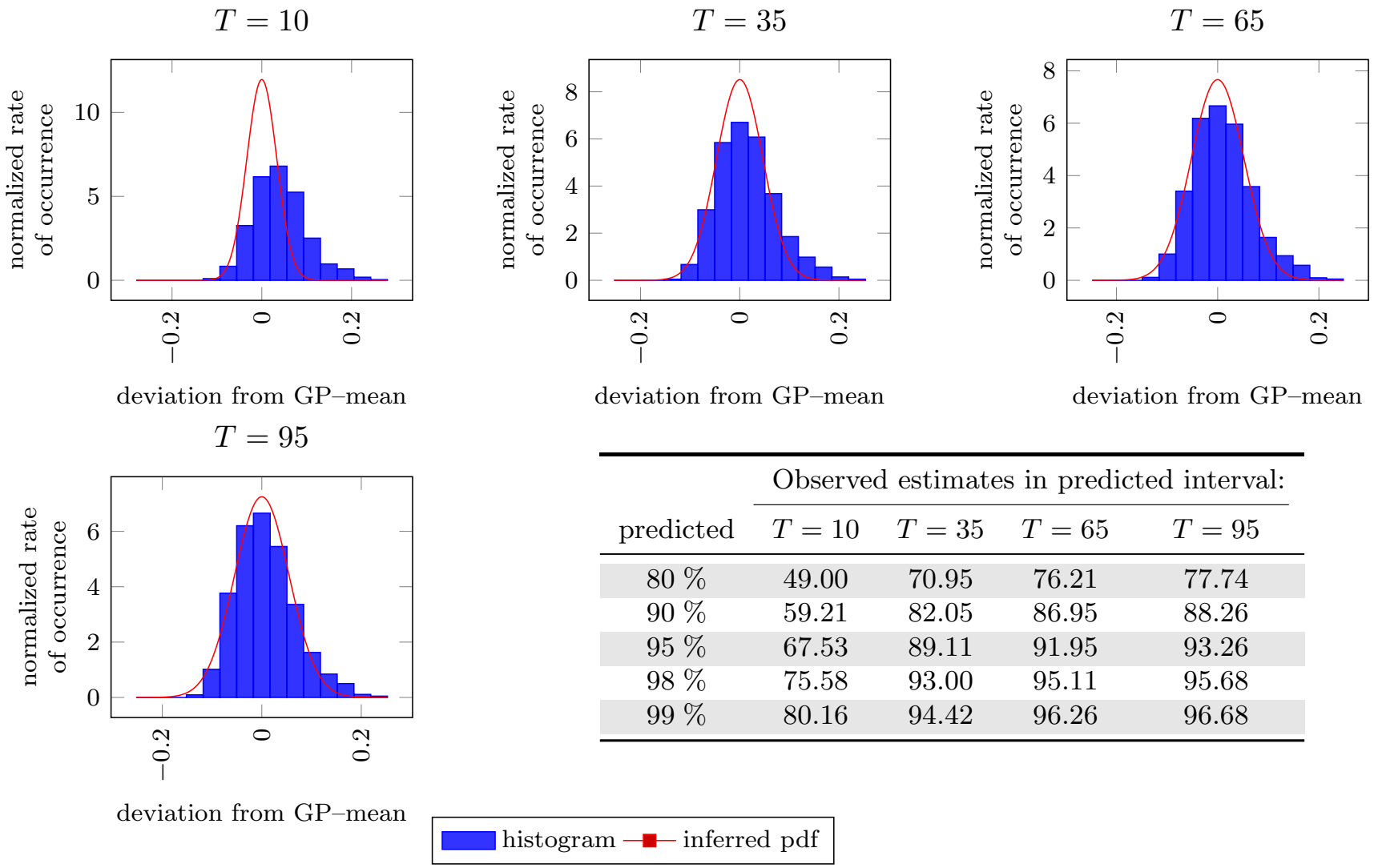




- Residual and error bound correlate with error



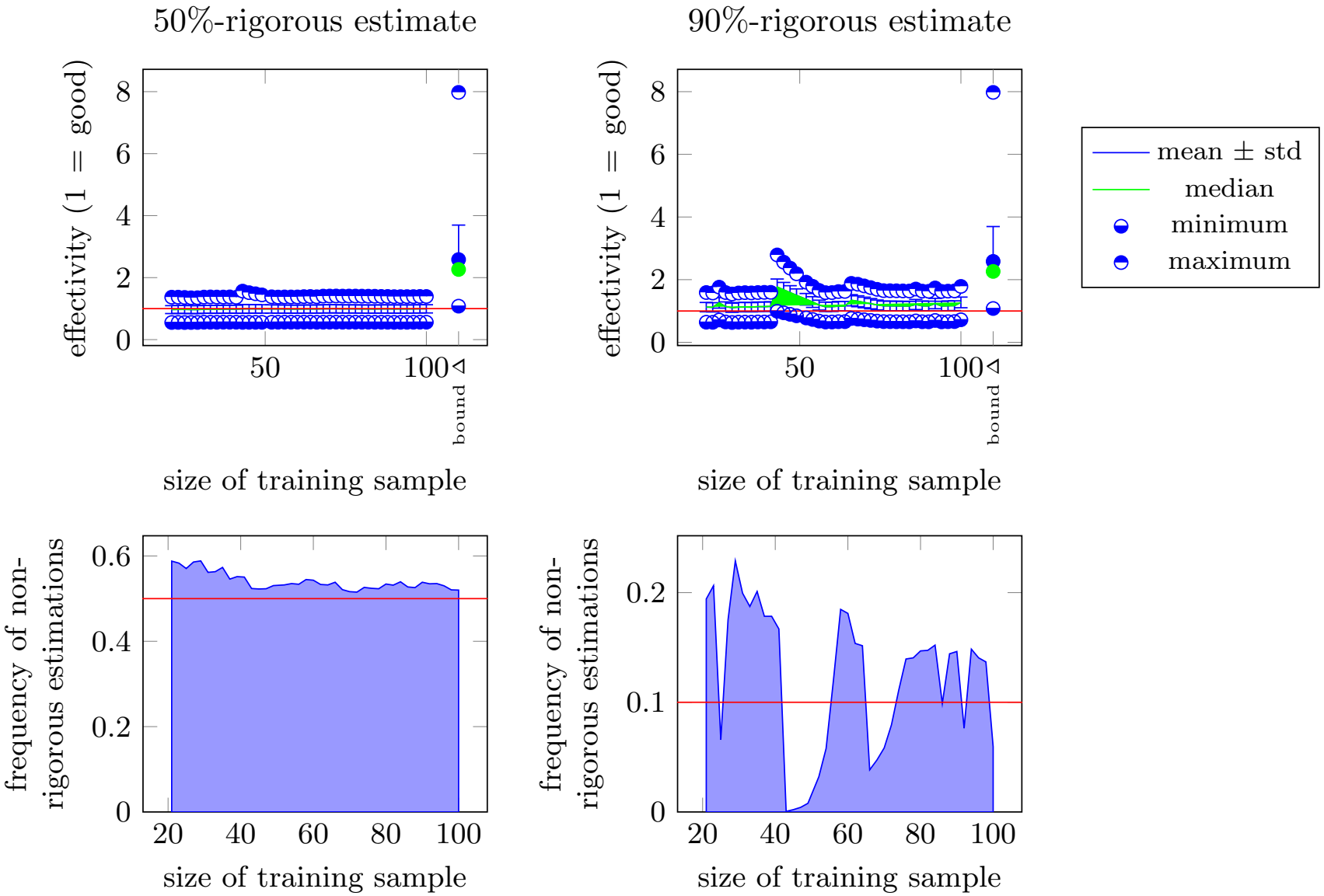

- ROMES (Kernel, residual indicator) in log-log space promising

# Gaussian-process assumptions verified (Kernel, residual indicator)

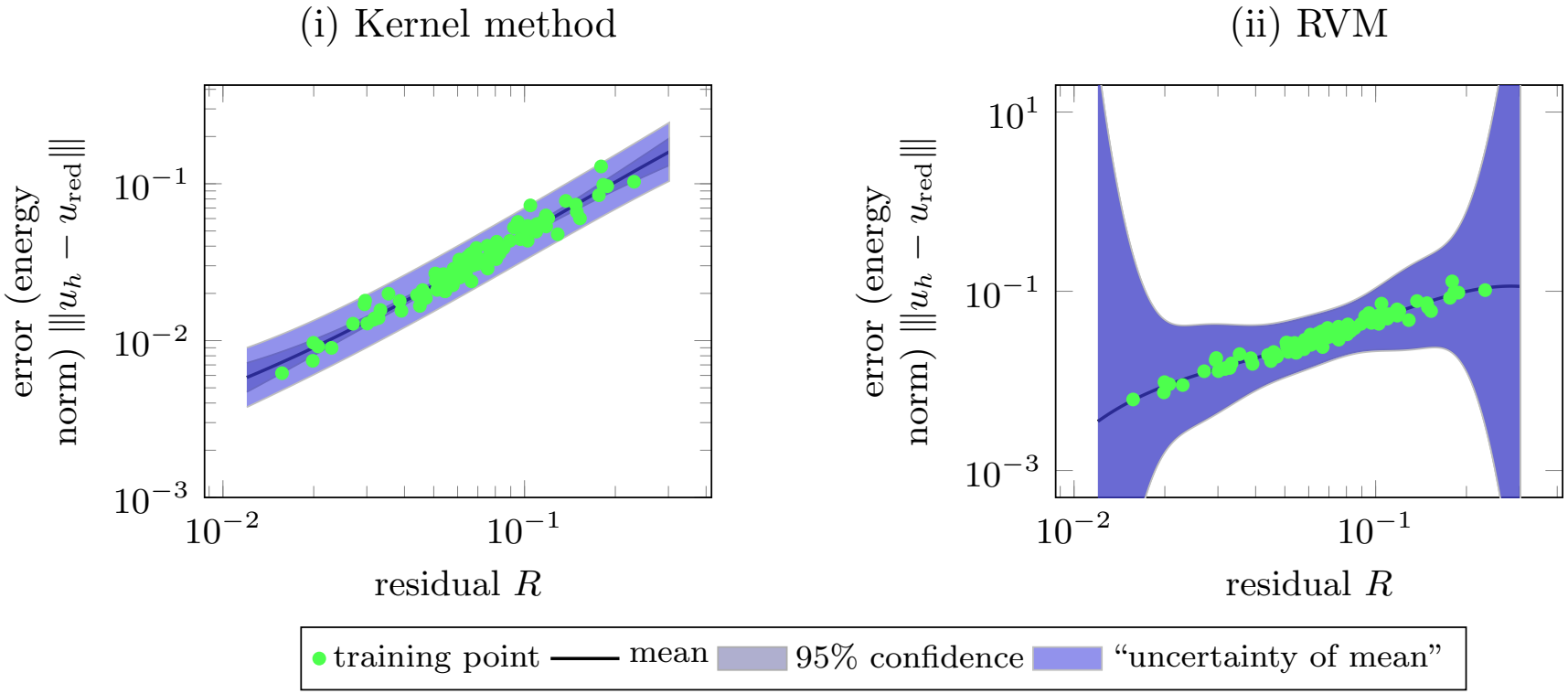



| predicted | Observed estimates in predicted interval: $T = 10$ | $T = 35$ | $T = 65$ | $T = 95$ |
|---|---|---|---|---|
| 80 % | 49.00 | 70.95 | 76.21 | 77.74 |
| 90 % | 59.21 | 82.05 | 86.95 | 88.26 |
| 95 % | 67.53 | 89.11 | 91.95 | 93.26 |
| 98 % | 75.58 | 93.00 | 95.11 | 95.68 |
| 99 % | 80.16 | 94.42 | 96.26 | 96.68 |

# Can achieve 'statistical rigor' (Kernel GP, residual indicator)

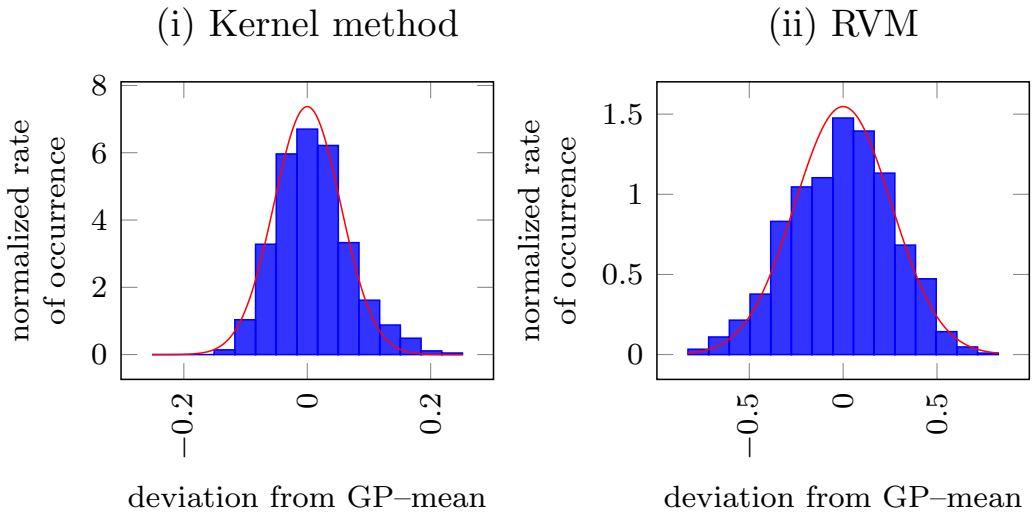


# ROMES: Kernel and RVM GP comparison



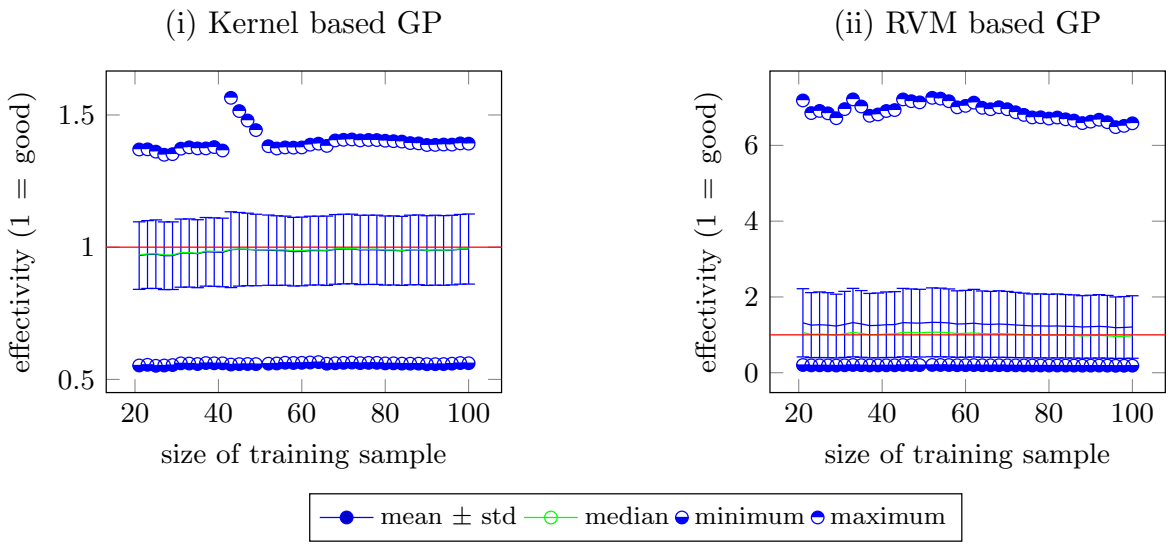

- RVM (Legendre-polynomial basis functions): significant uncertainty in mean's high-order polynomial coefficients

# ROMES: Kernel and RVM GP comparison




| $c$ | (i) | (ii) |
|---|---|---|
| 80 % | 77.7 | 77.9 |
| 90 % | 88.1 | 88.7 |
| 95 % | 92.9 | 94.7 |
| 98 % | 95.6 | 97.4 |
| 99 % | 96.6 | 99.2 |

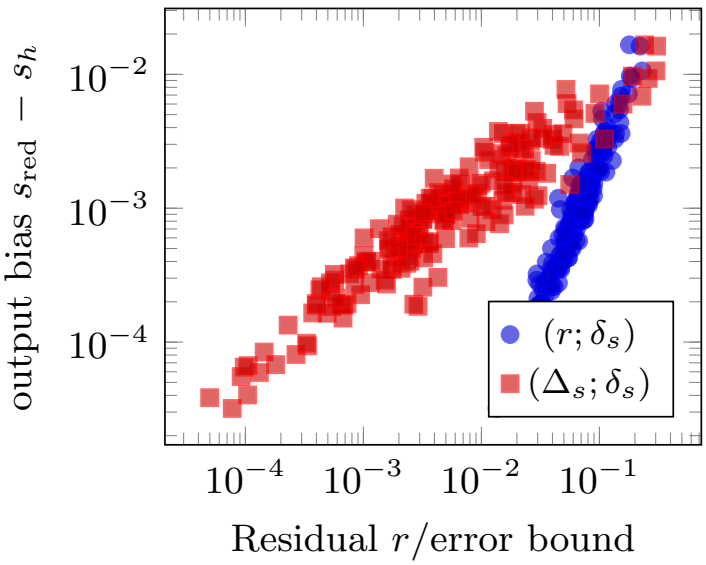
- GP structure and confidence intervals verified in both cases



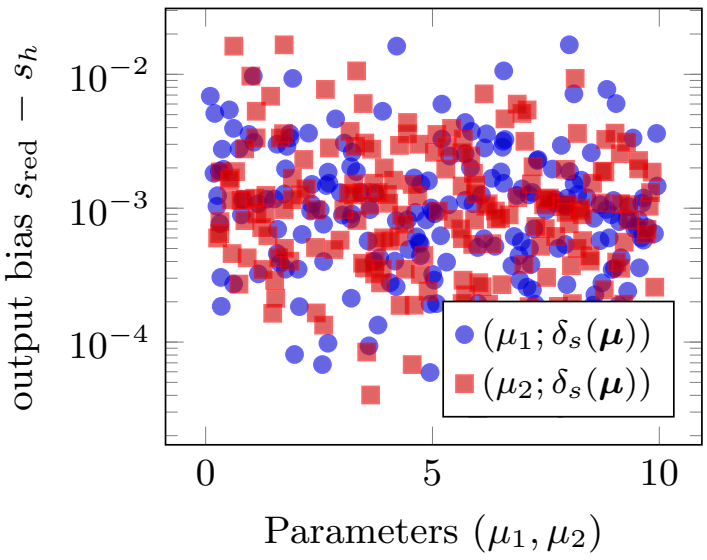

- Kernel GP produces better effectivity

# Error #2: output bias $\mathbf{z}(\boldsymbol{\mu}) - \mathbf{z}_{\text{red}}(\boldsymbol{\mu})$

(i) Indicators mapped to output bias



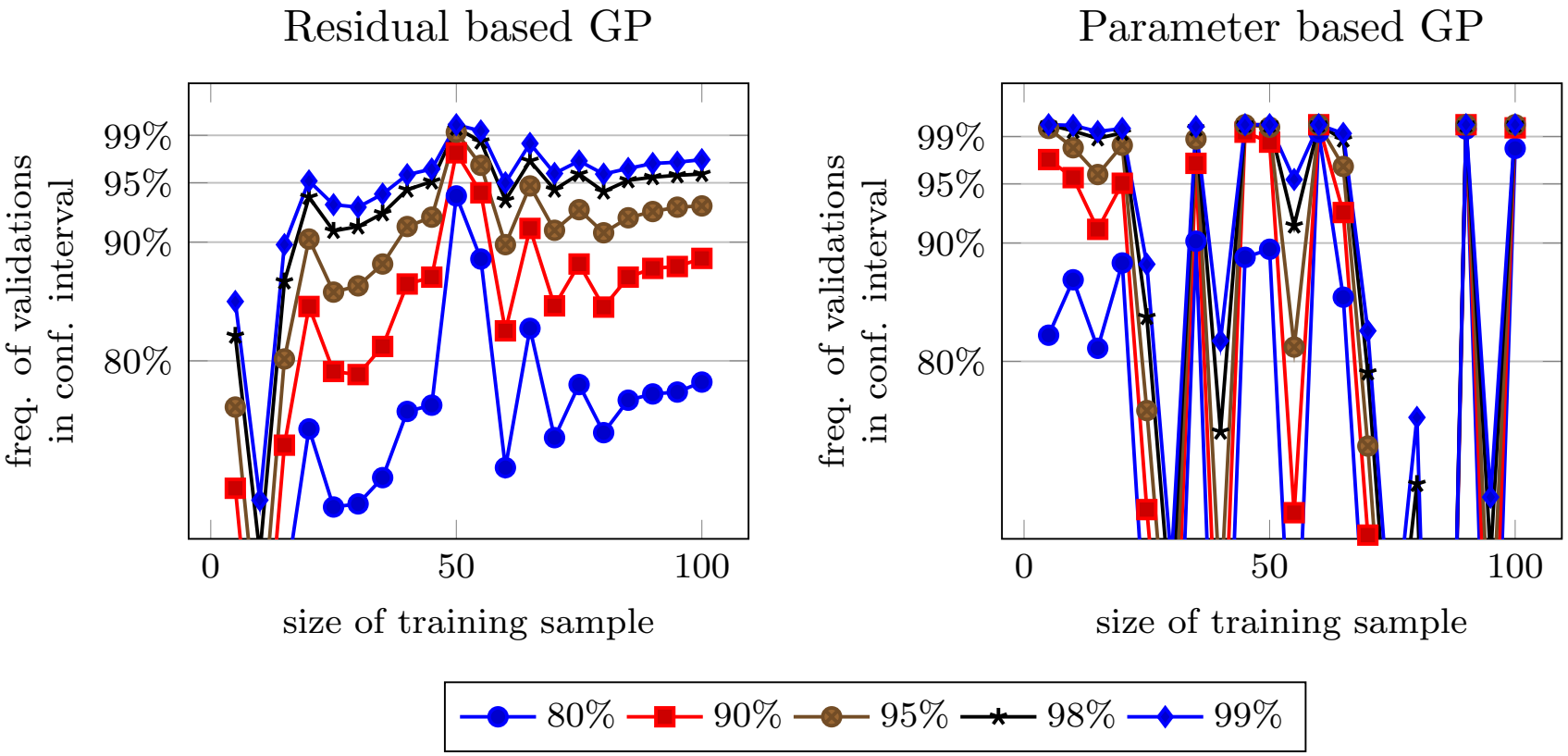

(ii) Parameters mapped to output bias




\+ Residual and error bound are good indicators (ROMES)

\- Inputs are poor indicators (Multifidelity correction)

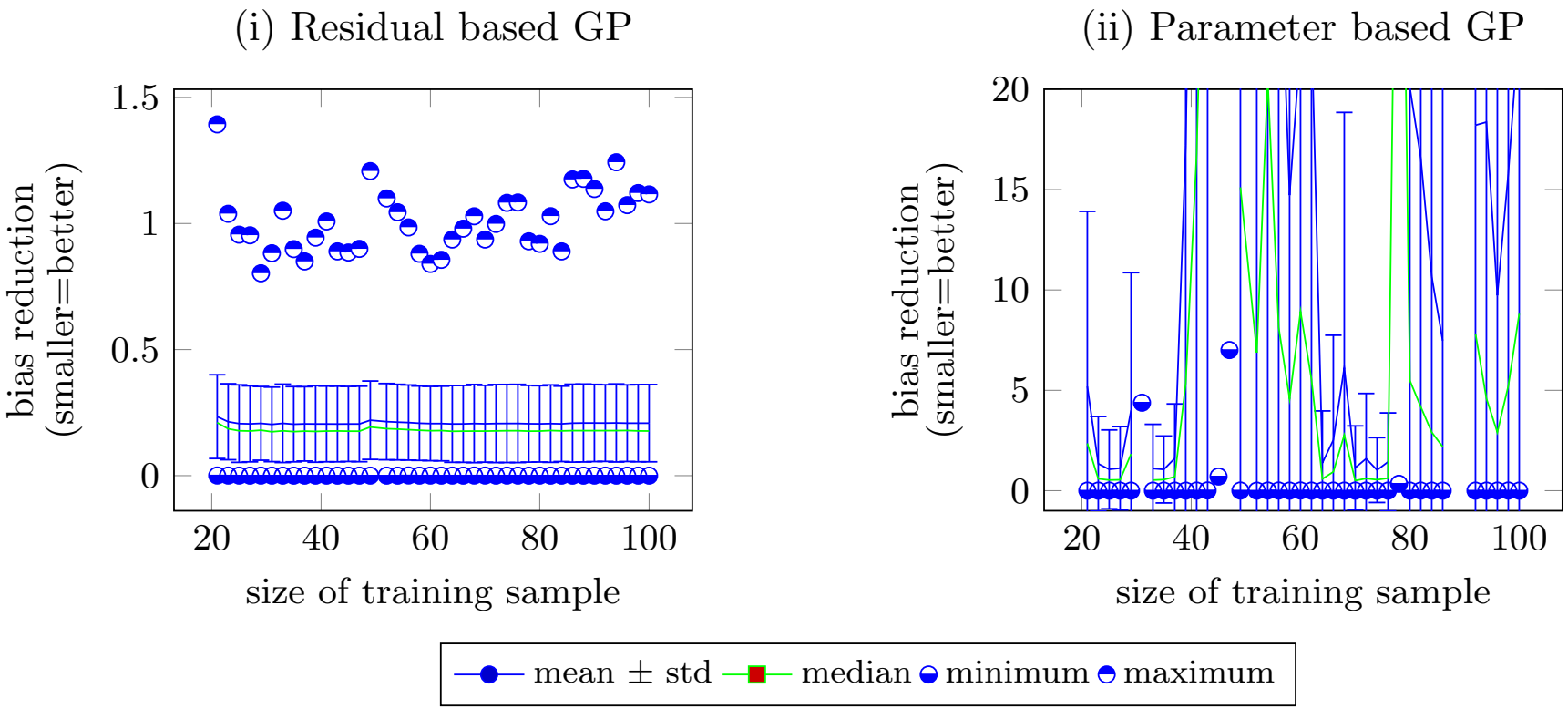
# Gaussian-process assumption verification




+ ROMES (Kernel GP, residual indicator) confidence intervals converge.
- Multifidelity correction (Kernel GP, input indicator) confidence intervals do not converge.

# Bias improvement

$$\text{bias reduction} = \frac{\mathsf{E}\left[\mathbf{z}_{\mathsf{red}}(\boldsymbol{\mu}) + \widetilde{\delta}_{\mathbf{z}}(\boldsymbol{\mu}) - \mathbf{z}(\boldsymbol{\mu})\right]}{\mathbf{z}_{\mathsf{red}}(\boldsymbol{\mu}) - \mathbf{z}(\boldsymbol{\mu})}$$




\+ ROMES reduces bias by roughly an order of magnitude

\- Multifidelity correction often *increases* the bias

# Conclusions

- ROMES
  - uses existing ROM error indicators to statistical quantify ROM error
  - 'statistical rigor' achievable
  - outperforms multifidelity correction (inputs = poor error indicators)
  - highlights strength of ROMs for data assimilation: other surrogates do not have such powerful error indicators
- Future work
  - apply to nonlinear, time-dependent problems
  - demonstrate on data-assimilation problem
  - automated selection of indicators and Gaussian process

# Conclusions

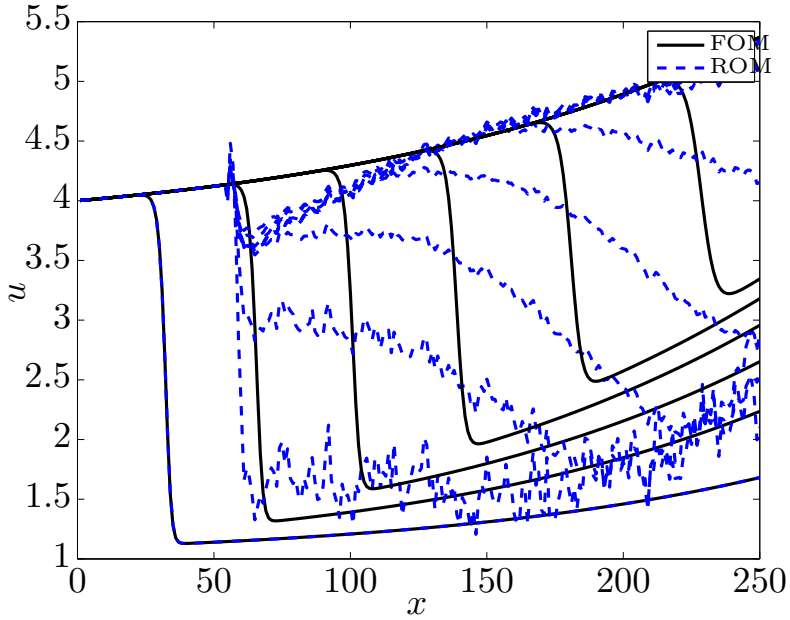
- ROMES
  - uses existing ROM error indicators to statistical quantify ROM error
  - 'statistical rigor' achievable
  - outperforms multifidelity correction (inputs = poor error indicators)
  - highlights strength of ROMs for data assimilation: other surrogates do not have such powerful error indicators
- Future work
  - apply to nonlinear, time-dependent problems
  - demonstrate on data-assimilation problem
  - automated selection of indicators and Gaussian process

# Acknowledgments

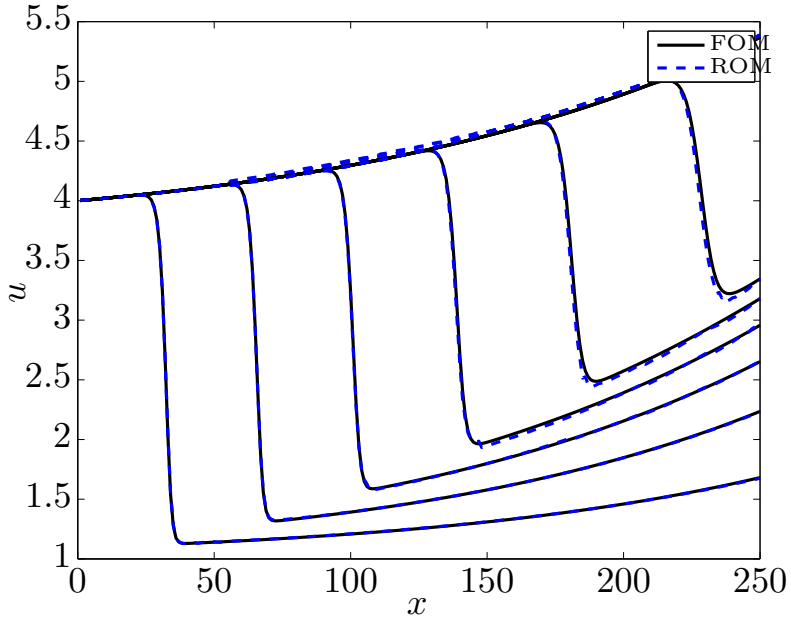

- Matthew Zahr: providing the model-reduction testbed that was modified to generate the numerical results
- Seshadhri Comandur, Khachik Sargsyan: helpful discussions
- This research was supported in part by an appointment to the Sandia National Laboratories Truman Fellowship in National Security Science and Engineering, sponsored by Sandia Corporation (a wholly owned subsidiary of Lockheed Martin Corporation) as Operator of Sandia National Laboratories under its U.S. Department of Energy Contract No. DE-AC04-94AL85000.

# Questions?

- **ROM error modeling**: M. Drohmann & K. Carlberg, "The ROMES method for statistical modeling of reduced-order-model error," arXiv e-print, (1404.0442), 2014.
- **ROM error control**: K. Carlberg. "Adaptive *h*-refinement for reduced-order models," International Journal for Numerical Methods in Engineering, in press (2014). doi:10.1002/nme.4800




(f) no adaptivity




(g) *h*-adaptivity

# Bibliography I


Amsallem, D., Cortial, J., Carlberg, K., and Farhat, C. (2009).
A method for interpolating on manifolds structural dynamics reduced-order models.
*International Journal for Numerical Methods in Engineering*, 80(9):1241–1258.

Amsallem, D. and Farhat, C. (2008).
An interpolation method for adapting reduced-order models and application to aeroelasticity.
*AIAA Journal*, 46(7):1803–1813.

Amsallem, D., Zahr, M. J., and Farhat, C. (2012).
Nonlinear model order reduction based on local reduced-order bases.
*International Journal for Numerical Methods in Engineering*, 92(10):891–916.

# Bibliography II


Arian, E., Fahl, M., and Sachs, E. W. (2000).
Trust-region proper orthogonal decomposition for flow control.
Technical Report 25, ICASE.

Babuška, I. and Miller, A. (1984).
The post-processing approach in the finite element method—part 1: Calculation of displacements, stresses and other higher derivatives of the displacements.
*International Journal for numerical methods in engineering*, 20(6):1085–1109.

Bangerth, W. and Rannacher, R. (1999).
Finite element approximation of the acoustic wave equation: Error control and mesh adaptation.
*East West Journal of Numerical Mathematics*, 7(4):263–282.

# Bibliography III


Bangerth, W. and Rannacher, R. (2003).
*Adaptive finite element methods for differential equations*.
Springer.

Becker, R. and Rannacher, R. (1996).
*Weighted a posteriori error control in finite element methods*, volume preprint no. 96-1.
Universitat Heidelberg.

Becker, R. and Rannacher, R. (2001).
An optimal control approach to a posteriori error estimation in finite element methods.
*Acta Numerica 2001*, 10:1–102.

# Bibliography IV


Carlberg, K. and Farhat, C. (2009).
An adaptive POD-Krylov reduced-order model for structural optimization.
*8th World Congress on Structural and Multidisciplinary Optimization, Lisbon, Portugal.*

Carlberg, K., Farhat, C., Cortial, J., and Amsallem, D. (2013).
The GNAT method for nonlinear model reduction: effective implementation and application to computational fluid dynamics and turbulent flows.
*Journal of Computational Physics*, 242:623–647.

Dihlmann, M., Drohmann, M., and Haasdonk, B.
Model reduction of parametrized evolution problems using the reduced basis method with adaptive time partitioning.

# Bibliography V


Drohmann, M., Haasdonk, B., and Ohlberger, M. (2011).
Adaptive reduced basis methods for nonlinear convection–diffusion equations.
In *Finite Volumes for Complex Applications VI Problems & Perspectives*, pages 369–377. Springer.

Eftang, J. L., Knezevic, D. J., and Patera, A. T. (2011).
An hp certified reduced basis method for parametrized parabolic partial differential equations.
*Mathematical and Computer Modelling of Dynamical Systems*, 17(4):395–422.

Eftang, J. L., Patera, A. T., and Rønquist, E. M. (2010).
An 'hp' certified reduced basis method for parametrized elliptic partial differential equations.
*SIAM Journal on Scientific Computing*, 32(6):3170–3200.

# Bibliography VI


Eldred, M. S., Giunta, A. A., Collis, S. S., Alexandrov, N. A., and Lewis, R. M. (2004).
Second-order corrections for surrogate-based optimization with model hierarchies.
In *Proceedings of the 10th AIAA/ISSMO Multidisciplinary Analysis and Optimization Conference, Albany, NY*, number AIAA Paper 2004-4457.

Eldred, M. S., Weickum, G., and Maute, K. (2009).
A multi-point reduced-order modeling approach of transient structural dynamics with application to robust design optimization.
*Structural and Multidisciplinary Optimization*, 38(6):599–611.

# Bibliography VII


Estep, D. (1995).
A posteriori error bounds and global error control for approximation of ordinary differential equations.
*SIAM Journal on Numerical Analysis*, 32(1):1–48.

Fidkowski, K. J. (2007).
*A simplex cut-cell adaptive method for high-order discretizations of the compressible Navier-Stokes equations*.
PhD thesis, Massachusetts Institute of Technology.

Galbally, D., Fidkowski, K., Willcox, K., and Ghattas, O. (2009).
Non-linear model reduction for uncertainty quantification in large-scale inverse problems.
*International Journal for Numerical Methods in Engineering*, 81(12):1581–1608.

# Bibliography VIII


Haasdonk, B., Dihlmann, M., and Ohlberger, M. (2011).
A training set and multiple bases generation approach for parameterized model reduction based on adaptive grids in parameter space.
*Mathematical and Computer Modelling of Dynamical Systems*, 17(4):423–442.

Huynh, D. B. P., Knezevic, D. J., Chen, Y., Hesthaven, J. S., and Patera, A. T. (2010).
A natural-norm successive constraint method for inf-sup lower bounds.
*Comput. Methods Appl. Mech. Engrg.*, 199:1963–1975.

# Bibliography IX


Lu, J. C.-C. (2005).
*An a posteriori error control framework for adaptive precision optimization using discontinuous Galerkin finite element method.*
PhD thesis, Massachusetts Institute of Technology.

Nemec, M. and Aftosmis, M. (2007).
Adjoint error estimation and adaptive refinement for embedded-boundary cartesian meshes.
In *18th AIAA CFD Conference, Miami. Paper AIAA-2007-4187.*

Ng, L. and Eldred, M. S. (2012).
Multifidelity uncertainty quantification using non-intrusive polynomial chaos and stochastic collocation.
In *AIAA 2012-1852.*

# Bibliography X


Park, M. A. (2004).
Adjoint-based, three-dimensional error prediction and grid adaptation.
*AIAA journal*, 42(9):1854–1862.

Patera, A. T. and Rozza, G. (2006).
*Reduced basis approximation and a posteriori error estimation for parametrized partial differential equations*.
MIT.

Peherstorfer, B., Butnaru, D., Willcox, K., and Bungartz, H.-J. (2014).
Localized discrete empirical interpolation method.
*SIAM Journal on Scientific Computing*, 36(1):A168—A192.

# Bibliography XI


Pierce, N. A. and Giles, M. B. (2000).
Adjoint recovery of superconvergent functionals from pde approximations.
*SIAM review*, 42(2):247–264.

Rannacher, R. (1999).
The dual-weighted-residual method for error control and mesh adaptation in finite element methods.
*MAFELEAP*, 99:97–115.

Rasmussen, C. and Williams, C. (2006).
*Gaussian Processes for Machine Learning*.
MIT Press.

# Bibliography XII


Rewienski, M. J. (2003).
*A Trajectory Piecewise-Linear Approach to Model Order Reduction of Nonlinear Dynamical Systems*.
PhD thesis, Massachusetts Institute of Technology.

Ryckelynck, D. (2005).
A priori hyperreduction method: an adaptive approach.
*Journal of Computational Physics*, 202(1):346–366.

Tipping, M. E. (2001).
Sparse bayesian learning and the relevance vector machine.
*The Journal of Machine Learning Research*, 1:211–244.

Venditti, D. and Darmofal, D. (2000).
Adjoint error estimation and grid adaptation for functional outputs: Application to quasi-one-dimensional flow.
*Journal of Computational Physics*, 164(1):204–227.

# Bibliography XIII


Venditti, D. A. and Darmofal, D. L. (2002).
Grid adaptation for functional outputs: application to two-dimensional inviscid flows.
*Journal of Computational Physics*, 176(1):40–69.

Washabaugh, K., Amsallem, D., Zahr, M., and Farhat, C. (2012).
Nonlinear model reduction for cfd problems using local reduced-order bases.
In *42nd AIAA Fluid Dynamics Conference and Exhibit, Fluid Dynamics and Co-located Conferences, AIAA Paper*, volume 2686.

# Bibliography XIV


Wirtz, D., Sorensen, D. C., and Haasdonk, B. (2012).
A-posteriori error estimation for DEIM reduced nonlinear dynamical systems.
*Preprint Series, Stuttgart Research Centre for Simulation Technology*.

Yano, M., Patera, A. T., and Urban, K. (2012).
A space-time certified reduced-basis method for Burgers' equation.
*Math. Mod. Meth. Appl. S.*, submitted.