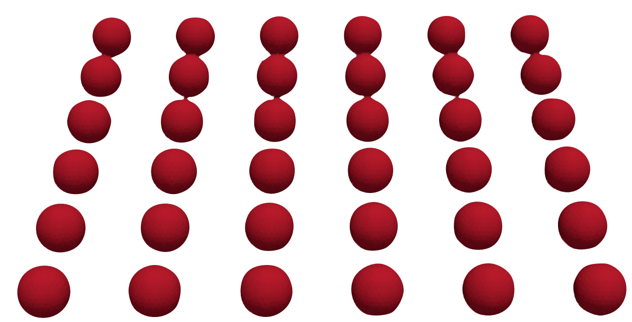
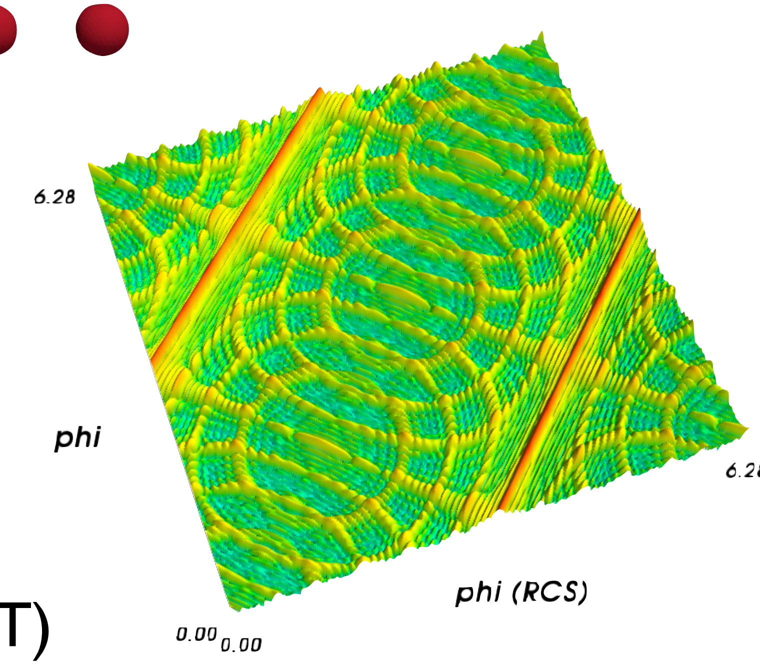
DTU 2016

Reduced order models for parameterized problems: Lecture Two

Jan S Hesthaven
EPFL-SB-MATHICSE-MCSS
jan.Hesthaven@epfl.ch





w/ assistance from B. Stamm (Aachen, D) and G. Rozza (SISSA, IT)

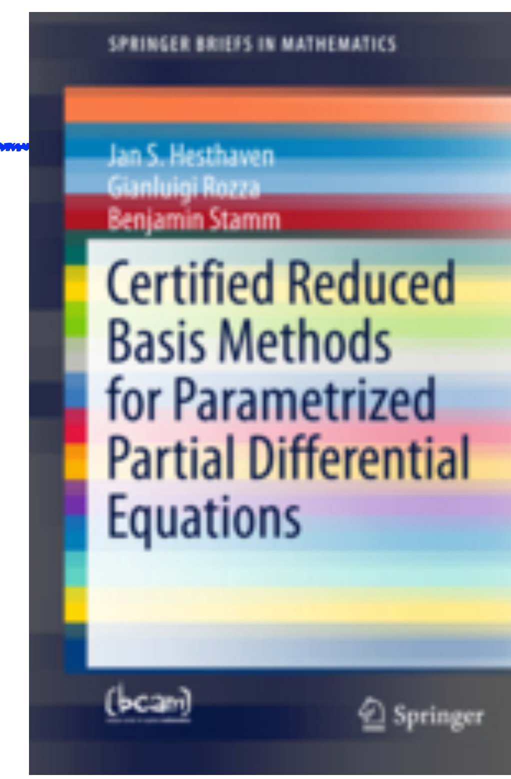
Overview of the lectures

Lecture 1: Introduction, motivation, basics

Lecture 2: Certified reduced methods

Lecture 3: The ‘ non’s ’ etc

Hesthaven, Rozza, Stamm
Certified Reduced Basis Methods for Parametrized Partial Differential Equations
Springer Briefs in Mathematics, 2015



Free: https://infoscience.epfl.ch/record/213266?ln=en

Overall goals

Understand Reduced models

Overall goals

Understand Reduced models

WHAT do we mean by 'reduced models' ?

WHY should we care ?

WHEN could it work ?

HOW do we know ?

DOES it work ?

WHAT's next ?

Starting point

We consider physical systems of the form

$$\mathcal{L}(\mathbf{x}, \mu)u(\mathbf{x}, \mu) = f(\mathbf{x}, \mu) \qquad \mathbf{x} \in \Omega$$

$$u(\mathbf{x}, \mu) = g(\mathbf{x}, \mu) \qquad \mathbf{x} \in \partial\Omega$$

where the solutions are implicitly parameterized by

$$\mu \in \mathcal{D} \subset \mathrm{R}^M$$

Starting point

We consider physical systems of the form

$$\mathcal{L}(\mathbf{x},\mu)u(\mathbf{x},\mu) = f(\mathbf{x},\mu) \qquad \mathbf{x} \in \Omega$$
$$u(\mathbf{x},\mu) = g(\mathbf{x},\mu) \qquad \mathbf{x} \in \partial\Omega$$

where the solutions are implicitly parameterized by

$$\mu \in \mathcal{D} \subset \mathrm{R}^M$$

- How do we find the basis.
- How do we ensure accuracy under parameter variation ?
- How do we ensure efficiency ?

The truth

Let us define:

The exact solution: Find $u(\mu) \in X$ such that

$$a(u, \mu, v) = f(\mu, v), \quad \forall v \in X$$

The truth

Let us define:

The exact solution: Find $u(\mu) \in X$ such that

$$a(u, \mu, v) = f(\mu, v), \quad \forall v \in X$$

The truth solution: Find $u_h(\mu) \in X_h$ such that

$$a_h(u_h, \mu, v_h) = f_h(\mu, v_h), \quad \forall v_h \in X_h \qquad \dim(X_h) = \mathcal{N}$$

The truth

Let us define:

The exact solution: Find $u(\mu) \in X$ such that

$$a(u, \mu, v) = f(\mu, v), \quad \forall v \in X$$

The truth solution: Find $u_h(\mu) \in X_h$ such that

$$a_h(u_h, \mu, v_h) = f_h(\mu, v_h), \quad \forall v_h \in X_h \qquad \dim(X_h) = \mathcal{N}$$

The RB solution: Find $u_{RB}(\mu) \in X_N$ such that

$$a_h(u_{RB}, \mu, v_N) = f_h(\mu, v_N), \forall v_N \in X_N \qquad \dim(X_N) = N$$

The truth

Let us define:

The exact solution: Find $u(\mu) \in X$ such that

$$a(u, \mu, v) = f(\mu, v), \quad \forall v \in X$$

The truth solution: Find $u_h(\mu) \in X_h$ such that

$$a_h(u_h, \mu, v_h) = f_h(\mu, v_h), \quad \forall v_h \in X_h \qquad \dim(X_h) = \mathcal{N}$$

The RB solution: Find $u_{RB}(\mu) \in X_N$ such that

$$a_h(u_{RB}, \mu, v_N) = f_h(\mu, v_N), \forall v_N \in X_N \qquad \dim(X_N) = N$$

We always assume that $\mathcal{N} \gg N$

The truth and errors

Solving for the truth is expensive - but we need to be able to trust the RB solution

$$\|u(\mu) - u_{RB}(\mu)\| \leq \|u(\mu) - u_h(\mu)\| + \|u_h(\mu) - u_{RB}(\mu)\|$$

The truth and errors

Solving for the truth is expensive - but we need to be able to trust the RB solution

$$\|u(\mu) - u_{RB}(\mu)\| \leq \|u(\mu) - u_h(\mu)\| + \|u_h(\mu) - u_{RB}(\mu)\|$$

We assume that

$$\|u(\mu) - u_h(\mu)\| \leq \varepsilon$$

This is <u>your favorite solver</u> and it is assumed it can be as accurate as you desire - the truth

The truth and errors

Solving for the truth is expensive - but we need to be able to trust the RB solution

$$\|u(\mu) - u_{RB}(\mu)\| \leq \|u(\mu) - u_h(\mu)\| + \|u_h(\mu) - u_{RB}(\mu)\|$$

We assume that

$$\|u(\mu) - u_h(\mu)\| \leq \varepsilon$$

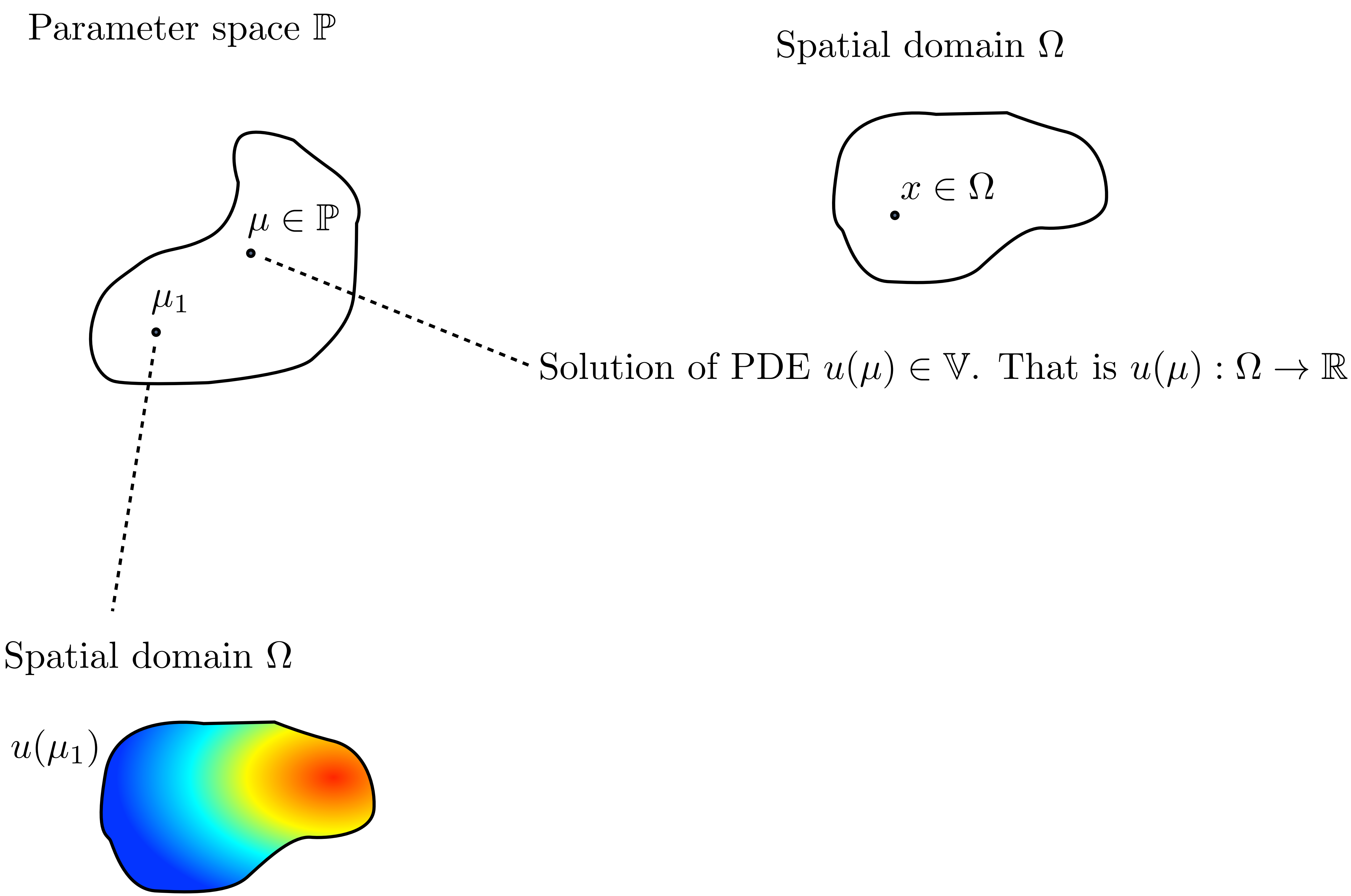
This is <u>your favorite solver</u> and it is assumed it can be as accurate as you desire - the truth

Bounding we achieve two things

- Ability to build a basis at minimal cost
- Certify the quality of the model

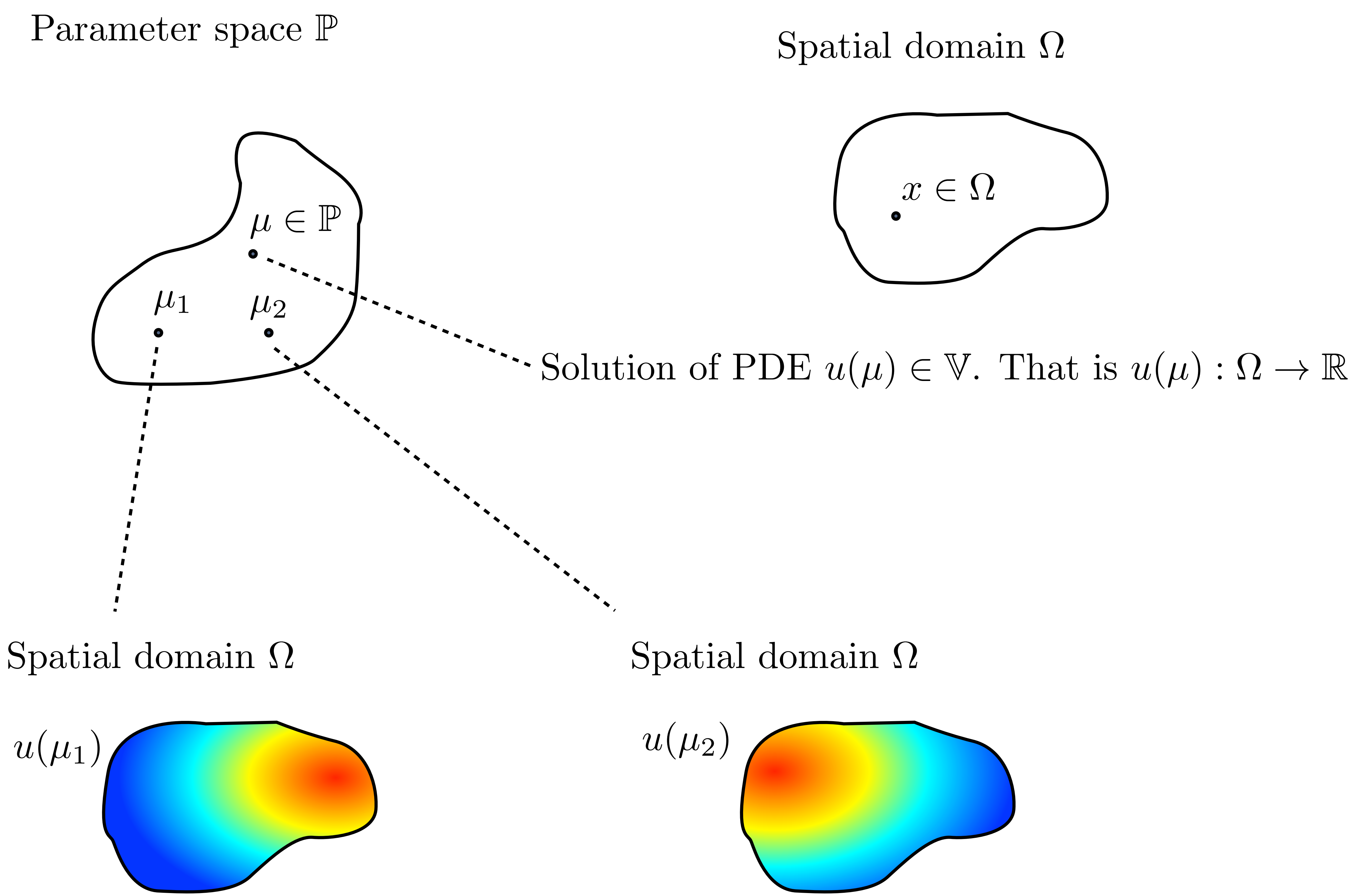
Greedy sampling

Notation: a function $v(\cdot;\mu) : \Omega \to \mathbb{R}$ is often denoted as $v(\mu)$.



Greedy sampling

Notation: a function $v(\cdot;\mu) : \Omega \to \mathbb{R}$ is often denoted as $v(\mu)$.



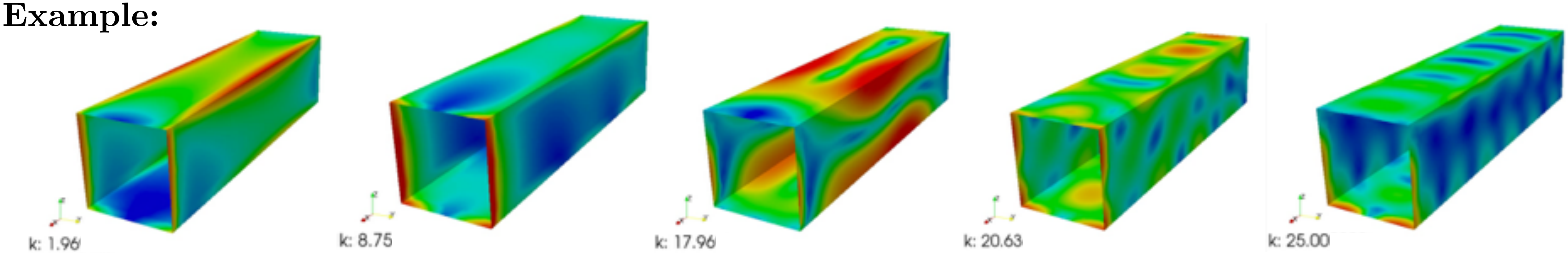
Greedy sampling

Reduced Basis Ansatz:

$$\mathbb{V}_{\texttt{rb}} = \text{span}\{u_\delta(\mu_1), \ldots, u_\delta(\mu_N)\}$$

for some well-chosen sample points $\mu_1, \ldots, \mu_N$.

Example:



1 parameter: wavenumber k
Different snapshots illustrated

Question: How to find the sample points $\mu_1, \ldots, \mu_N$ such that

$$\mathbb{V}_{\texttt{rb}} \approx \mathcal{M}_\delta = \{u_\delta(\mu) \,:\, \forall \mu \in \mathbb{P}\},$$

i.e. such that

$$E(\mathcal{M}_\delta, \mathbb{V}_{\texttt{rb}}) = \sup_{u_\delta(\mu) \in \mathcal{M}_\delta} \inf_{v_{\texttt{rb}} \in \mathbb{V}_{\texttt{rb}}} \|u_\delta(\mu) - v_{\texttt{rb}}\|_{\mathbb{V}} < \texttt{tol}$$

for N as little as possible?

Greedy sampling

Goal: Selection of sample points $\mu_1, \dots, \mu_N$ such that

$$\mathbb{V}_{\mathtt{rb}} = \text{span}\{u_\delta(\mu_1), \dots, u_\delta(\mu_N)\} \approx \mathcal{M}_\delta$$

Greedy algorithm:

Set $N = 1$, choose $\mu_1 \in \mathbb{P}$ arbitrarily.

1. Compute $u_\delta(\mu_N) \in \mathbb{V}_\delta$ (truth problem: computationally expensive)
2. Set $\mathbb{V}_{\mathtt{rb}} = \text{span}\{\mathbb{V}_{\mathtt{rb}}, u_\delta(\mu_N)\}$
3. Find $\mu_{N+1} = \arg\max_{\mu \in \mathbb{P}} \|u_\delta(\mu) - u_{\mathtt{rb}}(\mu)\|_{\mathbb{V}}$
4. Set $N := N + 1$ and `goto 1.` while $\max_{\mu \in \mathbb{P}} \|u_\delta(\mu) - u_{\mathtt{rb}}(\mu)\|_{\mathbb{V}} > \mathtt{Tol}$

Greedy sampling

ÉCOLE POLYTECHNIQUE FÉDÉRALE DE LAUSANNE

Goal: Selection of sample points $\mu_1, \ldots, \mu_N$ such that

$$\mathbb{V}_{\mathtt{rb}} = \mathrm{span}\{u_\delta(\mu_1), \ldots, u_\delta(\mu_N)\} \approx \mathcal{M}_\delta$$

Greedy algorithm:

Set $N = 1$, choose $\mu_1 \in \mathbb{P}$ arbitrarily.

1. Compute $u_\delta(\mu_N) \in \mathbb{V}_\delta$ (truth problem: computationally expensive)
2. Set $\mathbb{V}_{\mathtt{rb}} = \mathrm{span}\{\mathbb{V}_{\mathtt{rb}}, u_\delta(\mu_N)\}$
3. Find $\mu_{N+1} = \arg\max_{\mu \in \mathbb{P}} \|u_\delta(\mu) - u_{\mathtt{rb}}(\mu)\|_{\mathbb{V}}$
4. Set $N := N + 1$ and `goto 1.` while $\max_{\mu \in \mathbb{P}} \|u_\delta(\mu) - u_{\mathtt{rb}}(\mu)\|_{\mathbb{V}} > \mathtt{Tol}$

Remarks:

- In order to compute $\|u_\delta(\mu) - u_{\mathtt{rb}}(\mu)\|_{\mathbb{V}}$, the truth solution $u_\delta(\mu)$ needs to be computed.
- A sequence of hierarchical spaces is generated (which is not the case for the sequence of the best approximation spaces in the sens of Kolmogorov).

Greedy sampling

Key ingredient: (estimated) error feedback: Consider the mapping

$$\boxed{\mu \mapsto \eta(\mu),}$$

where $\eta(\mu)$ is an error estimation for $\|u_{\mathtt{rb}}(\mu) - u_\delta(\mu)\|_{\mathbb{V}}$.

Greedy sampling

Key ingredient: (estimated) error feedback: Consider the mapping

$$\boxed{\mu \mapsto \eta(\mu),}$$

where $\eta(\mu)$ is an error estimation for $\|u_{\mathtt{rb}}(\mu) - u_\delta(\mu)\|_{\mathbb{V}}$.

Recall that $u_\delta(\mu)$ the truth approximation defined by: find $u_\delta(\mu) \in \mathbb{V}_\delta$ such that

$$\boxed{a(u_\delta(\mu), v_\delta; \mu) = f(v_\delta; \mu), \ \forall v_\delta \in \mathbb{V}_\delta,}$$

and $u_{\mathtt{rb}}(\mu)$ the reduced basis solution defined by: find $u_{\mathtt{rb}}(\mu) \in \mathbb{V}_{\mathtt{rb}}$ such that

$$\boxed{a(u_{\mathtt{rb}}(\mu), v_{\mathtt{rb}}; \mu) = f(v_{\mathtt{rb}}; \mu), \ \forall v_{\mathtt{rb}} \in \mathbb{V}_{\mathtt{rb}}.}$$

The Galerkin framework allow for a residual-based a posteriori estimators without computing $u_\delta(\mu)$. This will be discussed at a later stage.

Greedy sampling

Greedy algorithm:

Set $N = 1$, choose $\mu_1 \in \mathbb{P}$ arbitrarily.

1. Compute $u_\delta(\mu_N) \in \mathbb{V}_\delta$ (truth problem: computationally expensive)
2. Set $\mathbb{V}_{\texttt{rb}} = \mathrm{span}\{\mathbb{V}_{\texttt{rb}}, u_\delta(\mu_N)\}$
3. Find $\quad \mu_{N+1} = \arg\max_{\mu\in\mathbb{P}} \eta(\mu)$
4. Set $N := N + 1$ and `goto 1.` while $\max_{\mu\in\mathbb{P}} \eta(\mu) > \texttt{tol}$

Greedy sampling

Greedy algorithm:

Set $N = 1$, choose $\mu_1 \in \mathbb{P}$ arbitrarily.

1. Compute $u_\delta(\mu_N) \in \mathbb{V}_\delta$ (truth problem: computationally expensive)
2. Set $\mathbb{V}_{\mathtt{rb}} = \mathrm{span}\{\mathbb{V}_{\mathtt{rb}}, u_\delta(\mu_N)\}$
3. Find $\mu_{N+1} = \arg\max_{\mu \in \mathbb{P}} \eta(\mu)$
4. Set $N := N + 1$ and `goto 1.` while $\max_{\mu \in \mathbb{P}} \eta(\mu) > \mathtt{tol}$

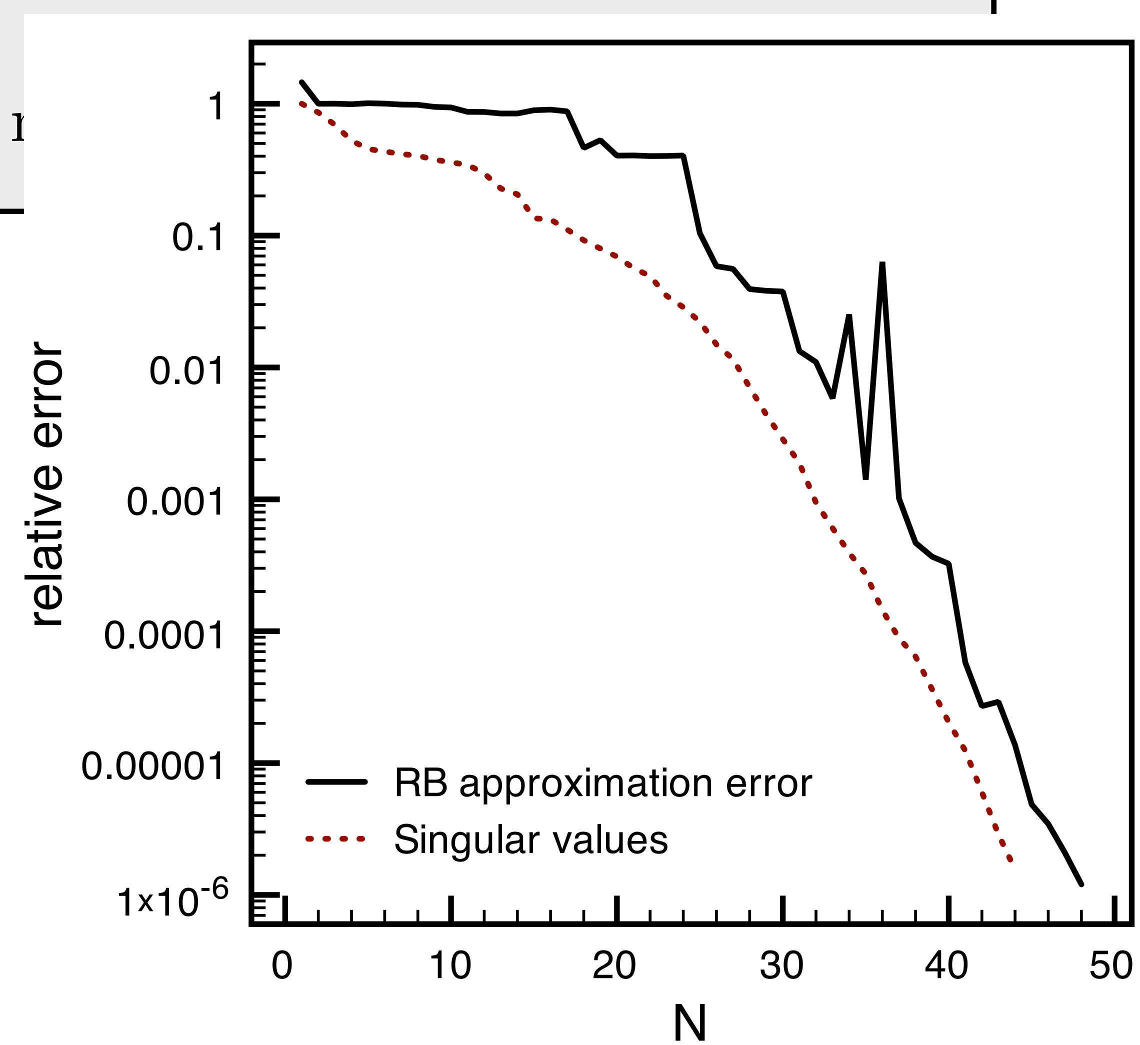
$$u_{RM}(\mu) = \sum_{i=1}^{N} u_N^i(\mu)\xi_i$$

Greedy sampling

Greedy algorithm:

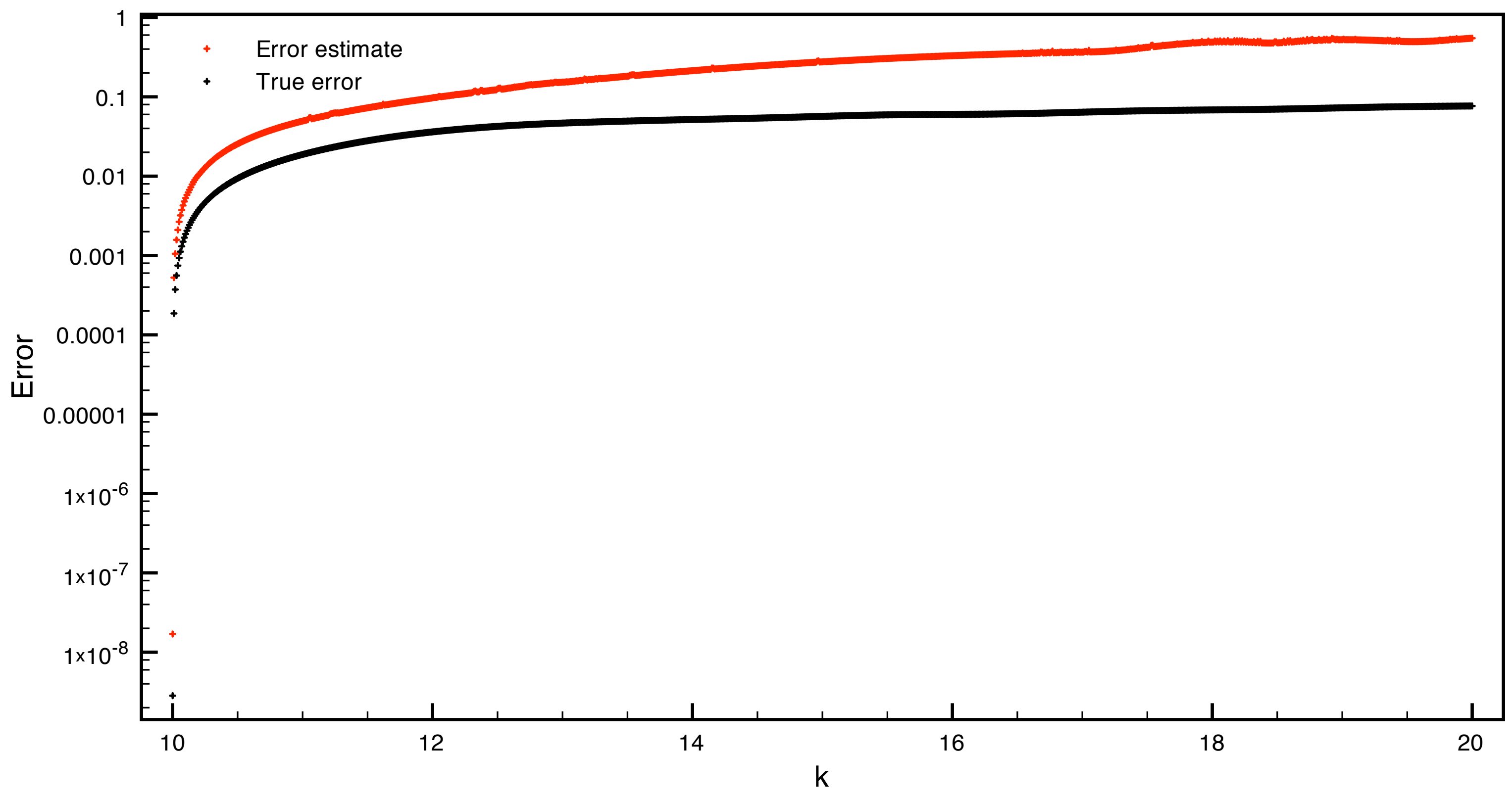
Set $N = 1$, choose $\mu_1 \in \mathbb{P}$ arbitrarily.

1. Compute $u_\delta(\mu_N) \in \mathbb{V}_\delta$ (truth problem: computationally expensive)
2. Set $\mathbb{V}_{\mathtt{rb}} = \mathrm{span}\{\mathbb{V}_{\mathtt{rb}}, u_\delta(\mu_N)\}$
3. Find $\quad \mu_{N+1} = \arg\max_{\mu \in \mathbb{P}} \eta(\mu)$
4. Set $N := N + 1$ and `goto 1.` while

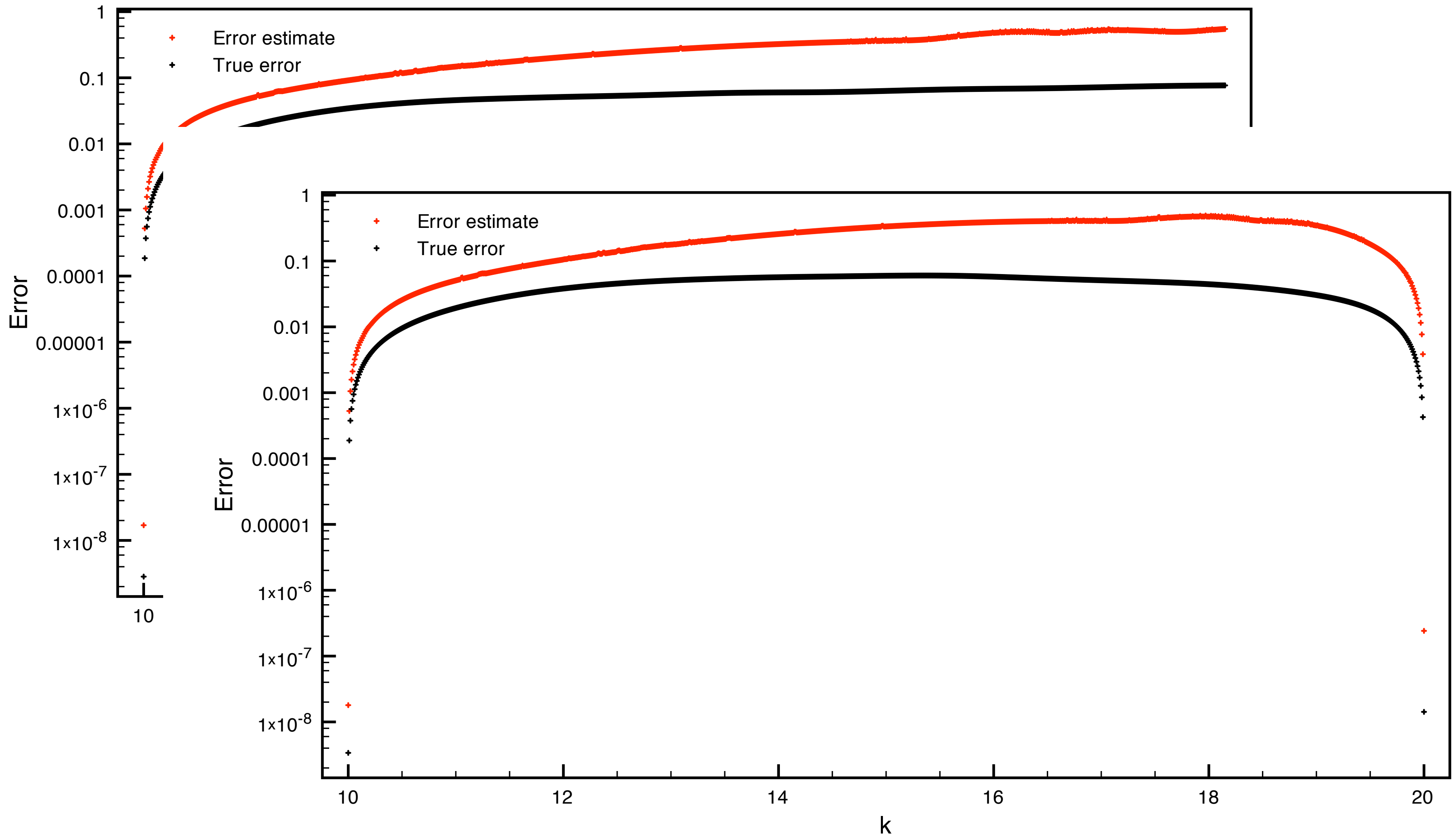


$$u_{RM}(\mu) = \sum_{i=1}^{N} u_N^i(\mu)\xi_i$$

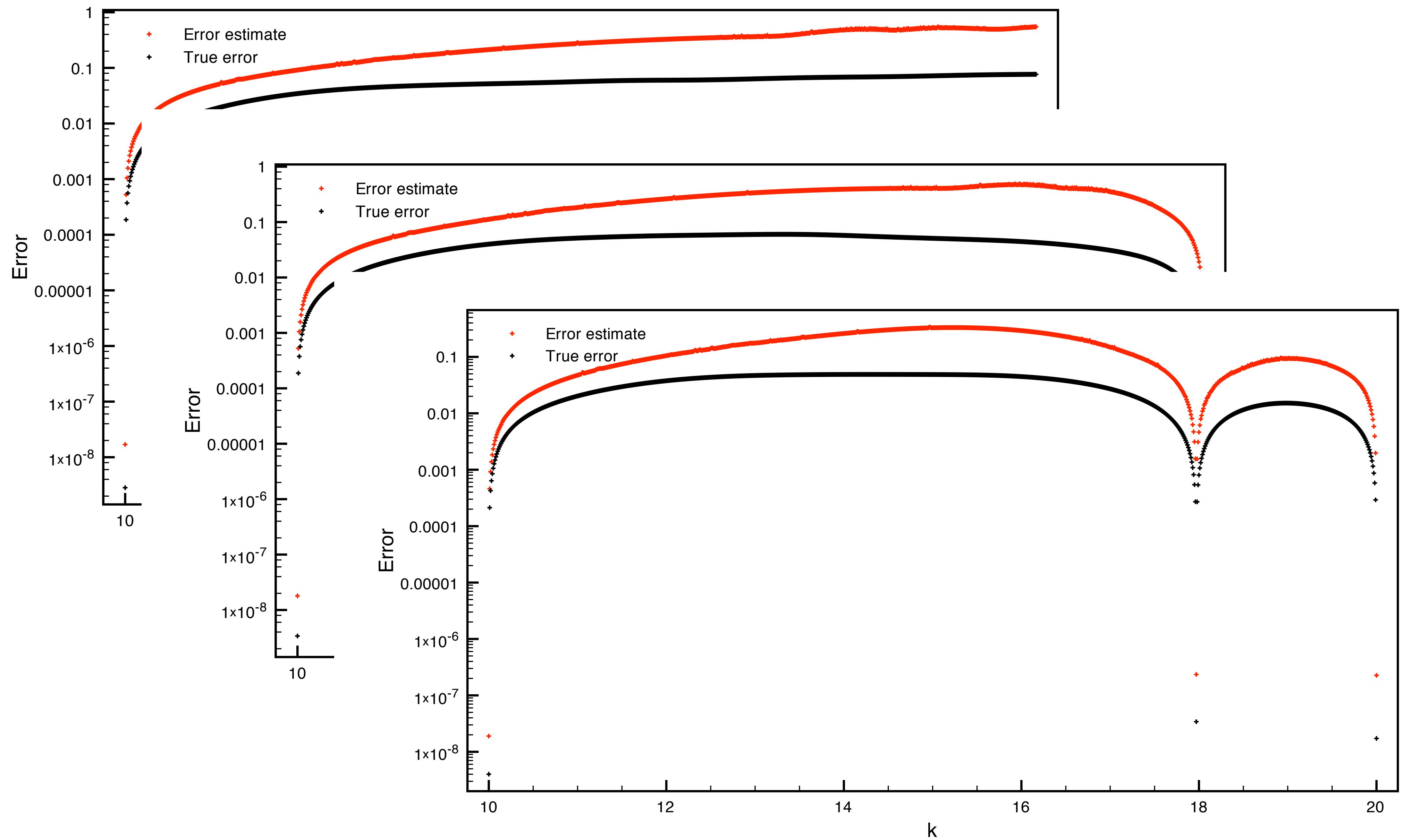
Greedy sampling - example



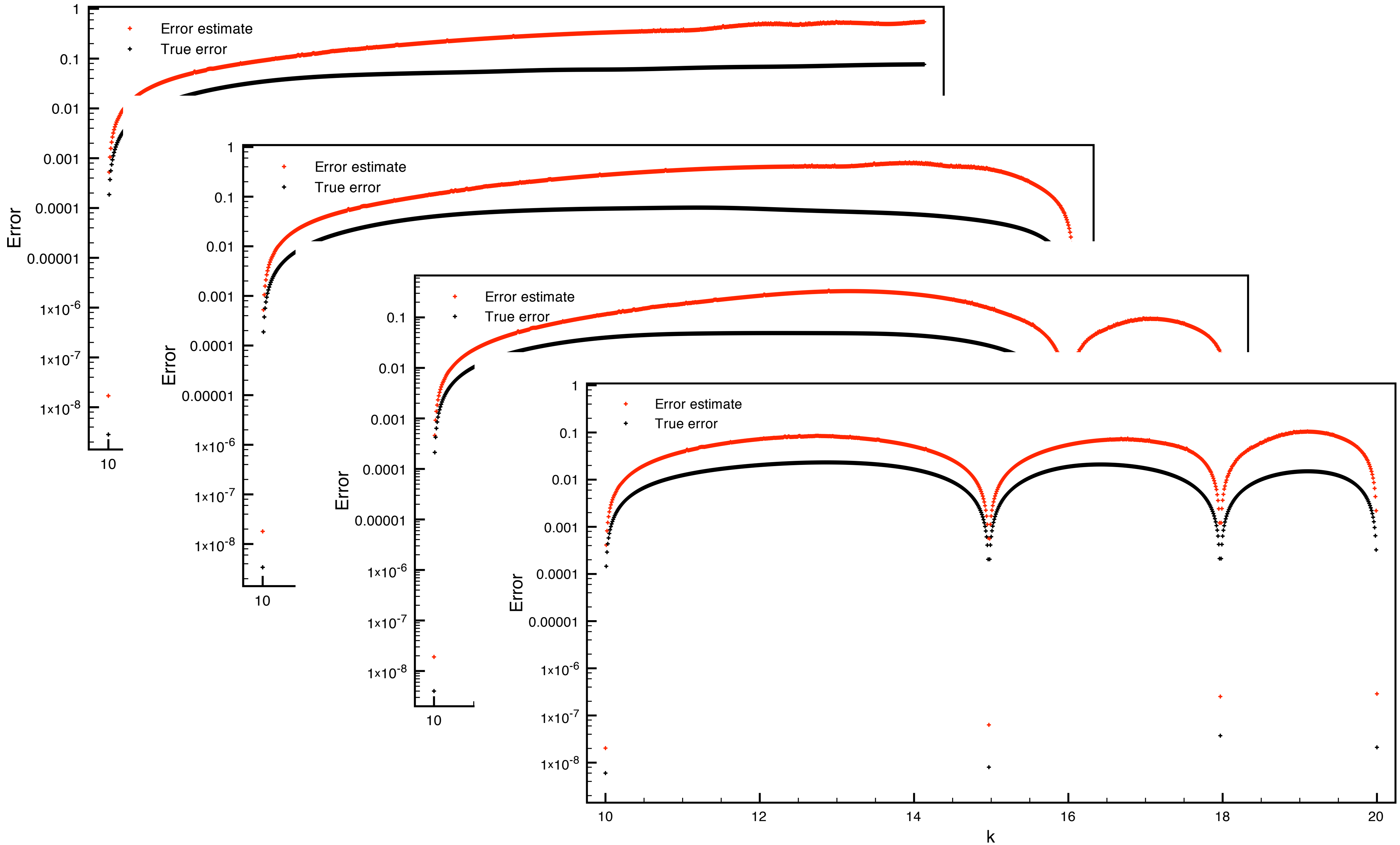
Greedy sampling - example



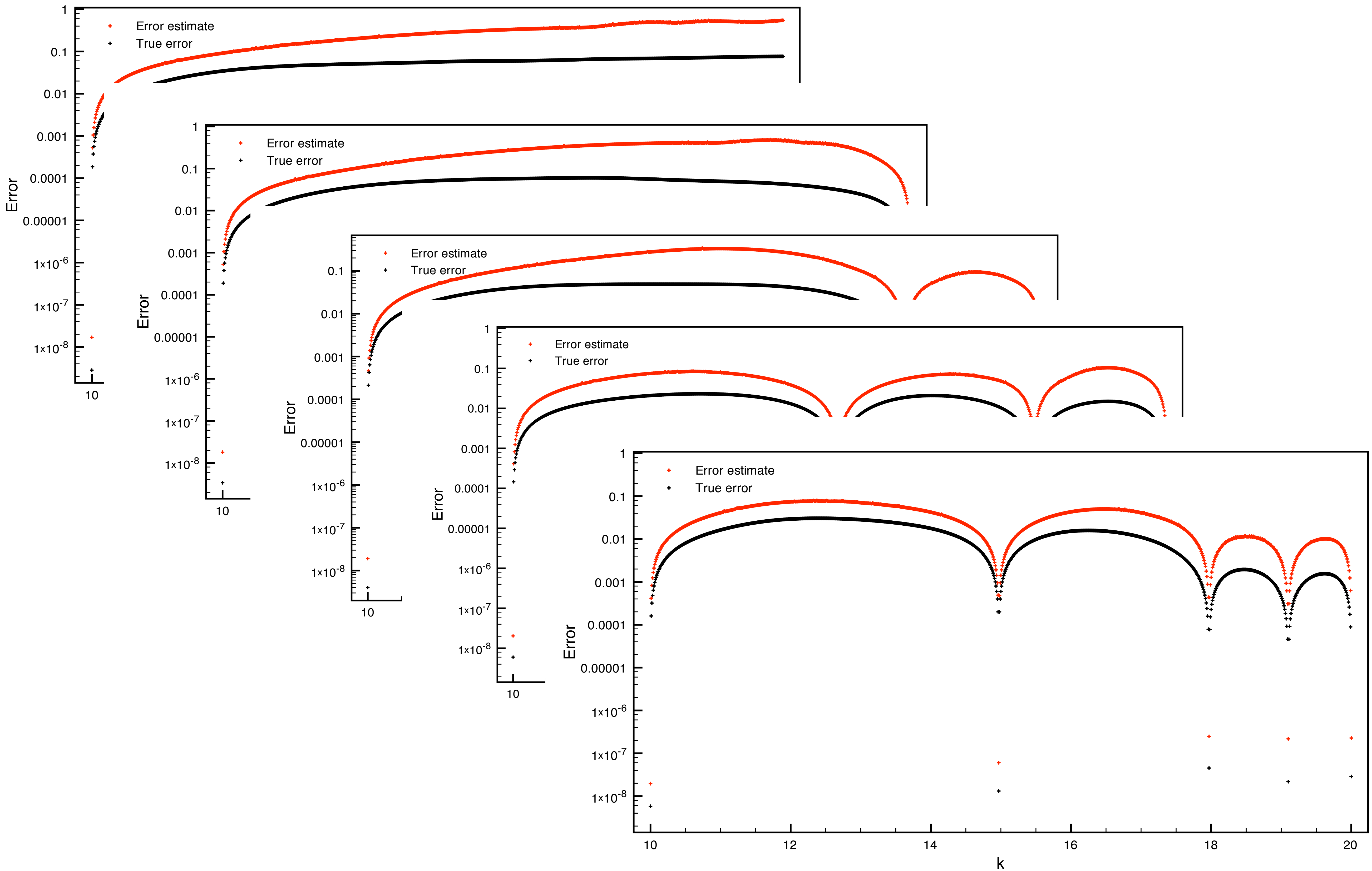
Greedy sampling - example



Greedy sampling - example



Greedy sampling - example



Greedy sampling - theory

Convergence: Is the convergence of the greedy algorithm comparable with the decay of the Kolmogorov N-width? Recall that the Kolmogorov N-width is the error using the best possible N-dimensional space (which is unknown in practise).

Theorem: Assume that the set of all solutions $\mathcal{M}$ has an exponentially small Kolmogorov N-width

$$\boxed{d_N(\mathcal{M}) \leq ce^{-aN},}$$

for an $a > \log\left(1 + \sqrt{\frac{\gamma}{\alpha}}\right)$, then the reduced basis approximation converges exponentially fast in the sense that there exists a $\beta > 0$ such that

$$\boxed{\forall \mu \in \mathbb{P} : \|u_\delta(\mu) - u_{\mathtt{rb}}(\mu)\|_{\mathbb{V}} \leq Ce^{-\beta N}.}$$

Another result exists which says that if the Kolmogorov N-width decays with an algebraic rate, then so does the reduced basis approximation based on the greedy-algorithm. The rates are however different.

Need for speed

How to solve the problem: For $\mu \in \mathbb{P}$, find the solution $u_{\texttt{rb}}(\mu) \in \mathbb{V}_{\texttt{rb}}$ of

$$\boxed{a(u_{\texttt{rb}}(\mu), v_{\texttt{rb}}; \mu) = f(v_{\texttt{rb}}; \mu), \qquad \forall v_{\texttt{rb}} \in \mathbb{V}_{\texttt{rb}}}$$

Need for speed

How to solve the problem: For $\mu \in \mathbb{P}$, find the solution $u_{\mathtt{rb}}(\mu) \in \mathbb{V}_{\mathtt{rb}}$ of

$$\boxed{a(u_{\mathtt{rb}}(\mu), v_{\mathtt{rb}}; \mu) = f(v_{\mathtt{rb}}; \mu), \qquad \forall v_{\mathtt{rb}} \in \mathbb{V}_{\mathtt{rb}}}$$

It consists of a linear problem with N degrees of freedom, thus results in a N-dimensional linear system

$$\boxed{\mathbf{A}^{\mu}_{\mathtt{rb}}\, \mathrm{u}^{\mu}_{\mathtt{rb}} = \mathrm{f}^{\mu}_{\mathtt{rb}},}$$

Need for speed

How to solve the problem: For $\mu \in \mathbb{P}$, find the solution $u_{\mathtt{rb}}(\mu) \in \mathbb{V}_{\mathtt{rb}}$ of

$$\boxed{a(u_{\mathtt{rb}}(\mu), v_{\mathtt{rb}}; \mu) = f(v_{\mathtt{rb}}; \mu), \qquad \forall v_{\mathtt{rb}} \in \mathbb{V}_{\mathtt{rb}}}$$

It consists of a linear problem with N degrees of freedom, thus results in a N-dimensional linear system

$$\boxed{\mathbf{A}^{\mu}_{\mathtt{rb}}\, \mathrm{u}^{\mu}_{\mathtt{rb}} = \mathrm{f}^{\mu}_{\mathtt{rb}},}$$

However: Let the reduced basis space be given by $\mathbb{V}_{\mathtt{rb}} = \mathrm{span}\{\xi_1, \ldots, \xi_N\}$, then the N-dimensional matrix

$$\boxed{(\mathbf{A}^{\mu}_{\mathtt{rb}})_{ij} = a(\xi_j, \xi_i; \mu), \quad 1 \leq i, j \leq N,}$$

needs to be reassembled for each new parameter value $\mu \in \mathbb{P}$ and the assembly process depends on $\mathcal{N}_\delta = \dim(\mathbb{V}_\delta)$: $\mathbf{A}^{\mu}_{\mathtt{rb}} = \mathbf{B}^T \mathbf{A}^{\mu}_{\delta} \mathbf{B}$.

Need for speed

The affine assumption

Assumption:

$$a(w,v;\mu) = \sum_{q=1}^{Q_{\mathtt{a}}} \theta_{\mathtt{a}}^q(\mu)\ a_q(w,v),$$

$$f(v;\mu) = \sum_{q=1}^{Q_{\mathtt{f}}} \theta_{\mathtt{f}}^q(\mu)\ f_q(v),$$

$$\ell(v;\mu) = \sum_{q=1}^{Q_{\mathtt{l}}} \theta_{\mathtt{l}}^q(\mu)\ \ell_q(v),$$

where

$$\theta_{\mathtt{a}}^q, \theta_{\mathtt{f}}^q, \theta_{\mathtt{l}}^q : \mathbb{P} \to \mathbb{R} \qquad \mu - \text{dependent functions},$$

$$a_q : \mathbb{V} \times \mathbb{V} \to \mathbb{R} \qquad \mu - \text{independent forms},$$

$$f_q, \ell_q : \mathbb{V} \to \mathbb{R} \qquad \mu - \text{independent forms},$$

Need for speed

Example: Convection-diffusion

$$a(u,v;\varepsilon) = \varepsilon \int_0^1 u'(x)v'(x)\,dx + \int_0^1 u'(x)v(x)\,dx,$$
$$a_1(u,v;\varepsilon) = \int_0^1 u'(x)v'(x)\,dx, \qquad \theta_{\mathsf{a}}^1(\varepsilon) = \varepsilon,$$
$$a_2(u,v;\varepsilon) = \int_0^1 u'(x)v(x)\,dx, \qquad \theta_{\mathsf{a}}^2(\varepsilon) = 1,$$

Example: Heat conduction on thermal blocks

$$a(w,v;\mu) = \sum_{i=1}^{15} \mu_i \int_{\mathcal{R}_i} \nabla w \cdot \nabla v + \int_{\mathcal{R}_{P+1}} \nabla w \cdot \nabla v,$$
$$a_i(w,v;\mu) = \int_{\mathcal{R}_i} \nabla w \cdot \nabla v, \qquad \theta_{\mathsf{a}}^i(\mu) = \mu_i, \qquad i = 1,\dots,15,$$
$$a_{16}(w,v;\mu) = \int_{\mathcal{R}_{16}} \nabla w \cdot \nabla v, \qquad \theta_{\mathsf{a}}^{16}(\mu) = 1,$$

Need for speed

Off-line: Given $\mathbb{V}_{\texttt{rb}} = \text{span}\{\xi_1, \ldots, \xi_N\}$ precompute

$$
\begin{aligned}
(\mathbf{A}^q_{\texttt{rb}})_{ij} &= a_q(\xi_j, \xi_i), & \forall\, 1 \leq i, j \leq N,\\
(\mathrm{f}^q_{\texttt{rb}})_i &= f_q(\xi_i), & \forall\, 1 \leq i \leq N,\\
(\mathrm{l}^q_{\texttt{rb}})_i &= \ell_q(\xi_i), & \forall\, 1 \leq i \leq N.
\end{aligned}
$$

Rem. Size of $\mathbf{A}^q_{\texttt{rb}}$ and $\mathrm{f}^q_{\texttt{rb}}$, $\mathrm{l}^q_{\texttt{rb}}$ is $N \times N$ resp. N.

Need for speed

Off-line: Given $\mathbb{V}_{\texttt{rb}} = \text{span}\{\xi_1, \ldots, \xi_N\}$ precompute

$$
\begin{aligned}
(\mathbf{A}^q_{\texttt{rb}})_{ij} &= a_q(\xi_j, \xi_i), && \forall\, 1 \le i, j \le N, \\
(\mathrm{f}^q_{\texttt{rb}})_i &= f_q(\xi_i), && \forall\, 1 \le i \le N, \\
(\mathrm{l}^q_{\texttt{rb}})_i &= \ell_q(\xi_i), && \forall\, 1 \le i \le N.
\end{aligned}
$$

Rem. Size of $\mathbf{A}^q_{\texttt{rb}}$ and $\mathrm{f}^q_{\texttt{rb}}$, $\mathrm{l}^q_{\texttt{rb}}$ is $N \times N$ resp. N.

Rem. The assembling depends on $N_\delta = \dim(\mathbb{V}_\delta)$. Indeed:

$$
\boxed{\mathbf{A}^q_{\texttt{rb}} = \mathbf{B}^T \mathbf{A}^q_\delta \mathbf{B}, \qquad \mathrm{f}^q_{\texttt{rb}} = \mathbf{B}^T \mathrm{f}^q_\delta, \qquad \mathrm{l}^q_{\texttt{rb}} = \mathbf{B}^T \mathrm{l}^q_\delta,}
$$

where $(\mathbf{A}^q_\delta)_{ij} = a_q(\varphi_j, \varphi_i)$, $(\mathrm{f}^q_\delta)_j = f_q(\varphi_j)$ and $(\mathrm{l}^q_\delta)_j = \ell_q(\varphi_j)$ for $1 \le i, j \le N_\delta$.

Need for speed

On-line: For each new parameter value $\mu \in \mathbb{P}$

1. Assemble (depending on $Q_{\mathtt{a}}, Q_{\mathtt{f}}$ and N, i.e. $\sim Q_{\mathtt{a}} N^2$ resp. $\sim Q_{\mathtt{f}} N$)

$$\mathbf{A}_{\mathtt{rb}}^{\mu} = \sum_{q=1}^{Q_{\mathtt{a}}} \theta_{\mathtt{a}}^{q}(\mu)\, \mathbf{A}_{\mathtt{rb}}^{q} \qquad \mathrm{f}_{\mathtt{rb}}^{\mu} = \sum_{q=1}^{Q_{\mathtt{f}}} \theta_{\mathtt{f}}^{q}(\mu)\, \mathrm{f}_{\mathtt{rb}}^{q}$$

2. Solve $\mathbf{A}_{\mathtt{rb}}^{\mu} \mathrm{u}_{\mathtt{rb}}^{\mu} = \mathrm{f}_{\mathtt{rb}}^{\mu}$. (depending on N, i.e $\sim N^3$ for LU factorization)
3. Compute

$$s_{\mathtt{rb}}(\mu) = \ell(u_{\mathtt{rb}}(\mu); \mu) = \sum_{q=1}^{Q_{\mathtt{l}}} \theta_{\mathtt{l}}^{m}(\mu)\; (\mathrm{u}_{\mathtt{rb}}^{\mu})^T\; \mathrm{l}_{\mathtt{rb}}^{q}.$$

Need for speed

On-line: For each new parameter value $\mu \in \mathbb{P}$

1. Assemble (depending on $Q_{\mathtt{a}}, Q_{\mathtt{f}}$ and N, i.e. $\sim Q_{\mathtt{a}} N^2$ resp. $\sim Q_{\mathtt{f}} N$)

$$\mathbf{A}^{\mu}_{\mathtt{rb}} = \sum_{q=1}^{Q_{\mathtt{a}}} \theta^q_{\mathtt{a}}(\mu)\, \mathbf{A}^q_{\mathtt{rb}} \qquad \mathrm{f}^{\mu}_{\mathtt{rb}} = \sum_{q=1}^{Q_{\mathtt{f}}} \theta^q_{\mathtt{f}}(\mu)\, \mathrm{f}^q_{\mathtt{rb}}$$

2. Solve $\mathbf{A}^{\mu}_{\mathtt{rb}} \mathrm{u}^{\mu}_{\mathtt{rb}} = \mathrm{f}^{\mu}_{\mathtt{rb}}$. (depending on N, i.e $\sim N^3$ for LU factorization)
3. Compute

$$s_{\mathtt{rb}}(\mu) = \ell(u_{\mathtt{rb}}(\mu); \mu) = \sum_{q=1}^{Q_{\mathtt{l}}} \theta^m_{\mathtt{l}}(\mu)\ (\mathrm{u}^{\mu}_{\mathtt{rb}})^T\ \mathrm{l}^q_{\mathtt{rb}}.$$

This is independent on N !

Need for speed

Computational costs:

- Off-line procedure: $T_{\texttt{off}}$.
- One on-line evaluation: $T_{\texttt{on}}$.
- One truth solve: $T_{\texttt{tr}} \gg T_{\texttt{on}}$.

Evaluation for M parameter values:

1. Brute force approach: $M \cdot T_{\texttt{tr}}$.
2. Reduced basis method: $T_{\texttt{off}} + M \cdot T_{\texttt{on}}$.

Need for speed

Computational costs:

- Off-line procedure: $T_{\texttt{off}}$.
- One on-line evaluation: $T_{\texttt{on}}$.
- One truth solve: $T_{\texttt{tr}} \gg T_{\texttt{on}}$.

Evaluation for M parameter values:

1. Brute force approach: $M \cdot T_{\texttt{tr}}$.
2. Reduced basis method: $T_{\texttt{off}} + M \cdot T_{\texttt{on}}$.

Theoretical considerations:

- The parameter space $\mathbb{P}$ is a continuous space (not discrete): M is potentially arbitrarily high.
- Whenever the number M of parameter evaluations is high enough, the reduced basis method is always cheaper.

Need for speed

Computational costs:

- Off-line procedure: $T_{\mathtt{off}}$.
- One on-line evaluation: $T_{\mathtt{on}}$.
- One truth solve: $T_{\mathtt{tr}} \gg T_{\mathtt{on}}$.

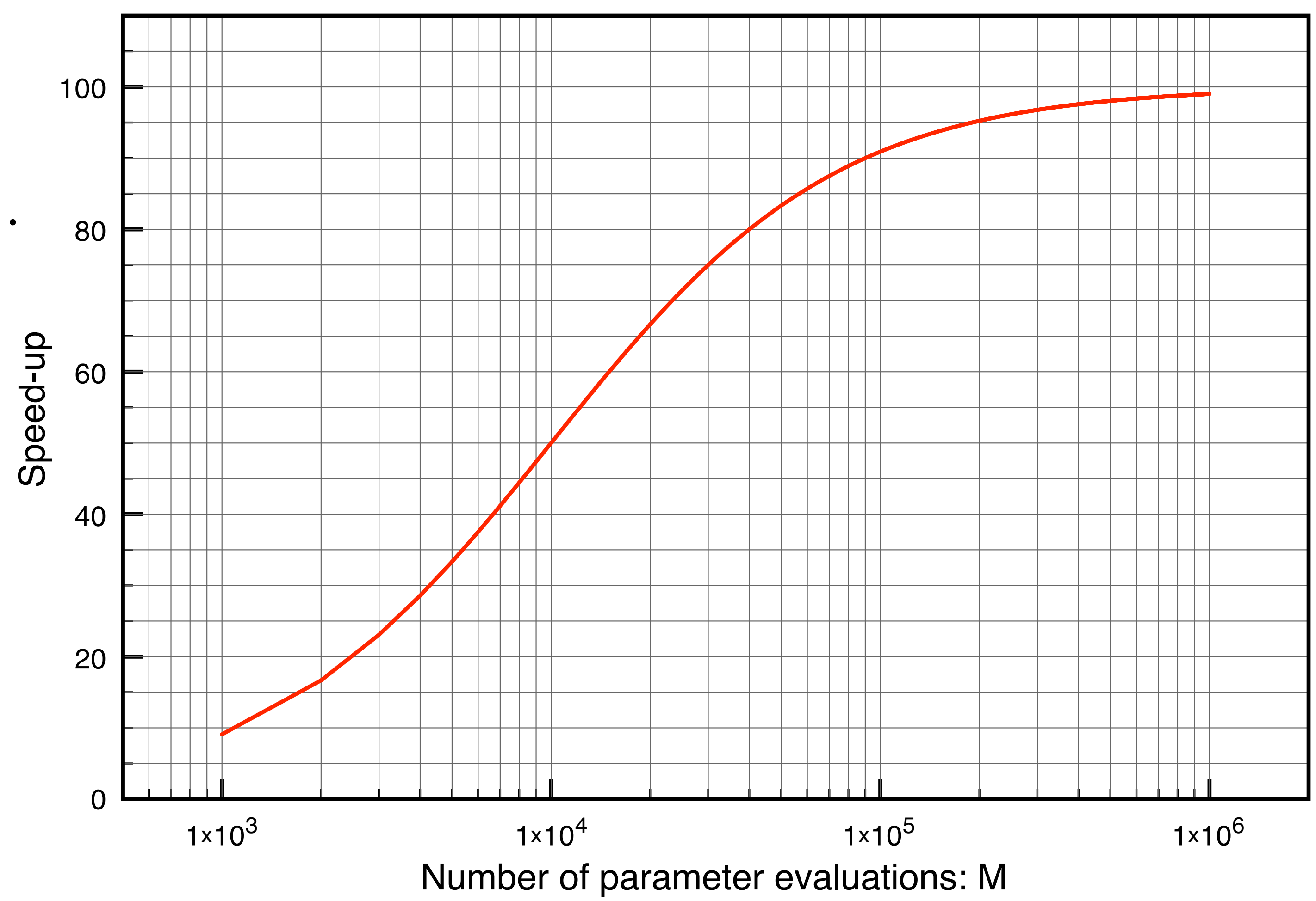
Evaluation for M parameter values:

1. Brute force approach: $M \cdot T_{\mathtt{tr}}$.
2. Reduced basis method: $T_{\mathtt{off}} + M \cdot T_{\mathtt{on}}$.

Realistic example:

- $T_{\mathtt{on}} = 1$.
- $T_{\mathtt{tr}} = 10^2$ (realistic speed-up).
- $T_{\mathtt{off}} = 10^4$ $(N = 100)$.

Brute force: $M \cdot T_{\mathtt{tr}} = M \cdot 10^2$
RBM: $T_{\mathtt{off}} + M \cdot T_{\mathtt{on}} = 10^4 + M$



The method - so far

Offline procedure:

1. Construct the reduced basis space $\mathbb{V}_{\texttt{rb}}$ empirically based on the weak greedy algorithm using an a posteriori estimator $\eta(\mu)$
2. Precompute the μ-independent matrices $\mathbf{A}^q_{\texttt{rb}}$ and the vectors $\mathrm{f}^q_{\texttt{rb}}, \mathrm{l}^q_{\texttt{rb}}$.

Online procedure:

$$\mu \longrightarrow \text{solve for: } u_{\texttt{rb}}(\mu) \longrightarrow s_{\texttt{rb}}(\mu) = \ell(u_{\texttt{rb}}(\mu); \mu)$$

which consists of

1. Assemble

$$\mathbf{A}^\mu_{\texttt{rb}} = \sum_{q=1}^{Q_{\texttt{a}}} \theta^q_{\texttt{a}}(\mu)\, \mathbf{A}^q_{\texttt{rb}} \qquad \mathrm{f}^\mu_{\texttt{rb}} = \sum_{q=1}^{Q_{\texttt{f}}} \theta^q_{\texttt{f}}(\mu)\, \mathrm{f}^q_{\texttt{rb}}$$

2. Solve $\mathbf{A}^\mu_{\texttt{rb}} \mathrm{u}^\mu_{\texttt{rb}} = \mathrm{f}^\mu_{\texttt{rb}}$ (N-dimensional linear system, $N \ll N_\delta$)
3. Compute $s_{\texttt{rb}}(\mu) = \ell(u_{\texttt{rb}}(\mu); \mu)$
4. Compute the a posteriori error estimator $\eta(\mu)$ to certify the accuracy

$$\|u_\delta(\mu) - u_{\texttt{rb}}(\mu)\|_{\mathbb{V}} \le \eta(\mu).$$

Characteristics: Independent of $N_\delta = \dim(\mathbb{V}_\delta)$: cheap. Feasible in a many-query context.

The method - so far

Offline procedure:

1. Construct the reduced basis space $\mathbb{V}_{\mathtt{rb}}$ empirically based on the weak greedy algorithm using an a posteriori estimator $\eta(\mu)$
2. Precompute the μ-independent matrices $\mathbf{A}^q_{\mathtt{rb}}$ and the vectors $\mathrm{f}^q_{\mathtt{rb}}, \mathrm{l}^q_{\mathtt{rb}}$.

Online procedure:

$$\mu \quad \longrightarrow \quad \text{solve for: } u_{\mathtt{rb}}(\mu) \quad \longrightarrow \quad s_{\mathtt{rb}}(\mu) = \ell(u_{\mathtt{rb}}(\mu); \mu)$$

which consists of

1. Assemble

$$\mathbf{A}^\mu_{\mathtt{rb}} = \sum_{q=1}^{Q_{\mathtt{a}}} \theta^q_{\mathtt{a}}(\mu)\, \mathbf{A}^q_{\mathtt{rb}} \qquad \mathrm{f}^\mu_{\mathtt{rb}} = \sum_{q=1}^{Q_{\mathtt{f}}} \theta^q_{\mathtt{f}}(\mu)\, \mathrm{f}^q_{\mathtt{rb}}$$

2. Solve $\mathbf{A}^\mu_{\mathtt{rb}} \mathrm{u}^\mu_{\mathtt{rb}} = \mathrm{f}^\mu_{\mathtt{rb}}$ (N-dimensional linear system, $N \ll N_\delta$)
3. Compute $s_{\mathtt{rb}}(\mu) = \ell(u_{\mathtt{rb}}(\mu); \mu)$
4. Compute the a posteriori error estimator $\eta(\mu)$ to certify the accuracy

$$\|u_\delta(\mu) - u_{\mathtt{rb}}(\mu)\|_{\mathbb{V}} \le \eta(\mu).$$

Characteristics: Independent of $N_\delta = \dim(\mathbb{V}_\delta)$: cheap. Feasible in a many-query context.

The a posteriori estimator

So far we assumed the existence of the *a posteriori* **estimation process:**

$$\mu \quad \longrightarrow \quad \text{solve: } a(u_{\mathtt{rb}}(\mu), v_{\mathtt{rb}}; \mu) = f(v_{\mathtt{rb}}; \mu), \ \forall v_{\mathtt{rb}} \in \mathbb{V}_{\mathtt{rb}} \quad \longrightarrow \quad \eta(\mu)$$

where $\eta(\mu)$ is an *a posteriori* estimation for $\|u_{\mathtt{rb}}(\mu) - u_\delta(\mu)\|_{\mathbb{V}_\delta}$.

- Estimates the discrete error $\|u_\delta(\mu) - u_{\mathtt{rb}}(\mu)\|_{\mathbb{V}}$ by $\eta(\mu)$: error with respect to the truth approximation $u_\delta(\mu)$.
- Ideal scenario:

$$\|u_\delta(\mu) - u_{\mathtt{rb}}(\mu)\|_{\mathbb{V}} \leq \eta(\mu), \qquad \forall \mu \in \mathbb{P}.$$

  Certifies the model order reduction error with a computable bound.
- Crucial for selection process in weak greedy algorithm. (Off-line)
- Should be cheap, i.e., independent on $N_\delta = \dim(\mathbb{V}_\delta)$. (Off- and On-line)

Need for speed

ÉCOLE POLYTECHNIQUE FÉDÉRALE DE LAUSANNE

Why only $\|u_\delta(\mu) - u_{\mathtt{rb}}(\mu)\|_{\mathbb{V}}$?

$$\|u(\mu) - u_{\mathtt{rb}}(\mu)\|_{\mathbb{V}} \leq \underbrace{\|u(\mu) - u_\delta(\mu)\|_{\mathbb{V}}}_{\text{Error of truth approximation}} + \underbrace{\|u_\delta(\mu) - u_{\mathtt{rb}}(\mu)\|_{\mathbb{V}}}_{\text{Error of model order reduction}}$$

Error of truth approximation ⇓ Given by the approximation space $\mathbb{V}_\delta$ ⇓ **Assumption:**

$$\sup_{\mu \in \mathbb{P}} \|u(\mu) - u_\delta(\mu)\|_{\mathbb{V}} \leq \mathtt{tol}$$

by the choice of δ in $\mathbb{V}_\delta$.

Error of model order reduction ⇓ Depends on the reduced basis space $\mathbb{V}_{\mathtt{rb}}$ ⇓ **Needs to be estimated**

The error estimate

Consider the discrete truth problem

$$\mathrm{A}(\mu)u_h(\mu) = f_h(\mu)$$

The error estimate

Consider the discrete truth problem

$$\mathrm{A}(\mu)u_h(\mu) = f_h(\mu)$$

Express the solution as

$$u_h = u_N + u_\perp$$

$$u_h \in X_h$$

$$u_N \in X_N$$

The error estimate

Consider the discrete truth problem

$$\mathrm{A}(\mu) u_h(\mu) = f_h(\mu)$$

Express the solution as

$$u_h = u_N + u_\perp \qquad \begin{array}{l} u_h \in X_h \\ u_N \in X_N \end{array}$$

This results in the truth problem

$$\begin{bmatrix} A_{1,1} & A_{1,2} \\ A_{2,1} & A_{2,2} \end{bmatrix} \begin{bmatrix} u_N \\ u_\perp \end{bmatrix} = \begin{bmatrix} f_{RB} \\ f_\perp \end{bmatrix}$$

as well as the reduced problem

$$A_{1,1} u_{RB} = f_{RB}$$

The error estimate

This yields the estimate for the error

$$\begin{bmatrix} A_{1,1} & A_{1,2} \\ A_{2,1} & A_{2,2} \end{bmatrix} \begin{bmatrix} u_N - u_{RB} \\ u_\perp \end{bmatrix} = \begin{bmatrix} 0 \\ f_\perp - A_{2,1} u_{RB} \end{bmatrix}$$

The error estimate

This yields the estimate for the error

$$\begin{bmatrix} A_{1,1} & A_{1,2} \\ A_{2,1} & A_{2,2} \end{bmatrix} \begin{bmatrix} u_N - u_{RB} \\ u_\perp \end{bmatrix} = \begin{bmatrix} 0 \\ f_\perp - A_{2,1} u_{RB} \end{bmatrix}$$

We can recognize the right hand side as

$$\begin{bmatrix} 0 \\ f_\perp - A_{2,1} u_{RB} \end{bmatrix} = \begin{bmatrix} f_{RB} - A_{1,1} u_{RB} \\ f_\perp - A_{2,1} u_{RB} \end{bmatrix} = f_h - A u_{RB} = R(\mu)$$

and we recover

$$\|u_h(\mu) - u_{RB}(\mu)\| \leq \|A^{-1}(\mu)\| \|R(\mu)\|$$

So with the residual and an estimate of the norm of the inverse of A we can bound the error

Error estimation

Truth solution: Find $u_\delta(\mu) \in \mathbb{V}_\delta$ such that

$$a(u_\delta(\mu), v_\delta; \mu) = f(v_\delta; \mu), \qquad \forall v_\delta \in \mathbb{V}_\delta.$$

RB solution: Find $u_{\mathtt{rb}}(\mu) \in \mathbb{V}_{\mathtt{rb}}$ such that

$$a(u_{\mathtt{rb}}(\mu), v_{\mathtt{rb}}; \mu) = f(v_{\mathtt{rb}}; \mu), \qquad \forall v_{\mathtt{rb}} \in \mathbb{V}_{\mathtt{rb}}.$$

Since $\mathbb{V}_{\mathtt{rb}} \subset \mathbb{V}_\delta$, there holds that

$$a(u_\delta(\mu) - u_{\mathtt{rb}}(\mu), v_\delta; \mu) = f(v_\delta; \mu) - a(u_{\mathtt{rb}}(\mu), v_\delta; \mu), \qquad \forall v_\delta \in \mathbb{V}_\delta.$$

Error estimation

Truth solution: Find $u_\delta(\mu) \in \mathbb{V}_\delta$ such that

$$a(u_\delta(\mu), v_\delta; \mu) = f(v_\delta; \mu), \qquad \forall v_\delta \in \mathbb{V}_\delta.$$

RB solution: Find $u_{\mathtt{rb}}(\mu) \in \mathbb{V}_{\mathtt{rb}}$ such that

$$a(u_{\mathtt{rb}}(\mu), v_{\mathtt{rb}}; \mu) = f(v_{\mathtt{rb}}; \mu), \qquad \forall v_{\mathtt{rb}} \in \mathbb{V}_{\mathtt{rb}}.$$

Since $\mathbb{V}_{\mathtt{rb}} \subset \mathbb{V}_\delta$, there holds that

$$a(u_\delta(\mu) - u_{\mathtt{rb}}(\mu), v_\delta; \mu) = f(v_\delta; \mu) - a(u_{\mathtt{rb}}(\mu), v_\delta; \mu), \qquad \forall v_\delta \in \mathbb{V}_\delta.$$

Define the error function $e(\mu) \in \mathbb{V}_\delta$ and the residual $r(\cdot; \mu) \in \mathbb{V}'_\delta$ by

$$\begin{aligned} e(\mu) &= u_\delta(\mu) - u_{\mathtt{rb}}(\mu) \in \mathbb{V}_\delta, \\ r(v_\delta; \mu) &= f(v_\delta; \mu) - a(u_{\mathtt{rb}}(\mu), v_\delta; \mu), \qquad \forall v_\delta \in \mathbb{V}_\delta. \end{aligned}$$

This establishes the following error equation

$$a(e(\mu), v_\delta; \mu) = r(v_\delta; \mu), \qquad \forall v_\delta \in \mathbb{V}_\delta.$$

Error estimation

For coercive problems:

$$\alpha_\delta(\mu)\,\|v_\delta\|_{\mathbb{V}}^2 \le a(v_\delta, v_\delta; \mu),$$

with $\alpha_\delta(\mu) \ge \alpha_\delta > 0$.

In consequence

$$\begin{aligned}\|\underbrace{u_\delta(\mu) - u_{\mathtt{rb}}(\mu)}_{=e(\mu)}\|_{\mathbb{V}}^2 &\le \frac{1}{\alpha_\delta(\mu)} a(e(\mu), e(\mu); \mu) = \frac{1}{\alpha_\delta(\mu)} r(e(\mu); \mu)\\ &= \frac{1}{\alpha_\delta(\mu)} (\hat{r}_\delta(\mu), e(\mu))_{\mathbb{V}} \le \frac{1}{\alpha_\delta(\mu)} \|\hat{r}_\delta(\mu)\|_{\mathbb{V}}\ \|e(\mu)\|_{\mathbb{V}}\end{aligned}$$

Thus

$$\|u_\delta(\mu) - u_{\mathtt{rb}}(\mu)\|_{\mathbb{V}} \le \frac{\|\hat{r}_\delta(\mu)\|_{\mathbb{V}}}{\alpha_\delta(\mu)}$$

Error estimation

For coercive problems:

$$\alpha_\delta(\mu)\,\|v_\delta\|_{\mathbb{V}}^2 \leq a(v_\delta, v_\delta; \mu),$$

with $\alpha_\delta(\mu) \geq \alpha_\delta > 0$.

In consequence

$$\|\underbrace{u_\delta(\mu) - u_{\mathtt{rb}}(\mu)}_{=e(\mu)}\|_{\mathbb{V}}^2 \leq \frac{1}{\alpha_\delta(\mu)} a(e(\mu), e(\mu); \mu) = \frac{1}{\alpha_\delta(\mu)} r(e(\mu); \mu)$$
$$= \frac{1}{\alpha_\delta(\mu)} (\hat{r}_\delta(\mu), e(\mu))_{\mathbb{V}} \leq \frac{1}{\alpha_\delta(\mu)} \|\hat{r}_\delta(\mu)\|_{\mathbb{V}}\ \|e(\mu)\|_{\mathbb{V}}$$

Thus

$$\|u_\delta(\mu) - u_{\mathtt{rb}}(\mu)\|_{\mathbb{V}} \leq \frac{\|\hat{r}_\delta(\mu)\|_{\mathbb{V}}}{\alpha_\delta(\mu)}$$

Computing the stability constant

In the coercive case $\alpha_\delta(\mu)$ is defined by

$$\alpha_\delta(\mu) = \inf_{v_\delta \in \mathbb{V}_\delta} \frac{a(v_\delta, v_\delta; \mu)}{\|v_\delta\|^2_{\mathbb{V}}} = \inf_{v_\delta \in \mathbb{V}_\delta} \frac{a(v_\delta, v_\delta; \mu)}{(v_\delta, v_\delta)_{\mathbb{V}}}.$$

Thus $\alpha_\delta(\mu)$ is the smallest eigenvalue of: find $(\lambda, w_\delta) \in \mathbb{R}^+ \times \mathbb{V}_\delta$ such that

$$a(w_\delta, v_\delta; \mu) = \lambda \ (w_\delta, v_\delta)_{\mathbb{V}}, \qquad \forall v_\delta \in \mathbb{V}_\delta.$$

Computing the stability constant

In the coercive case $\alpha_\delta(\mu)$ is defined by

$$\alpha_\delta(\mu) = \inf_{v_\delta \in \mathbb{V}_\delta} \frac{a(v_\delta, v_\delta; \mu)}{\|v_\delta\|_{\mathbb{V}}^2} = \inf_{v_\delta \in \mathbb{V}_\delta} \frac{a(v_\delta, v_\delta; \mu)}{(v_\delta, v_\delta)_{\mathbb{V}}}.$$

Thus $\alpha_\delta(\mu)$ is the smallest eigenvalue of: find $(\lambda, w_\delta) \in \mathbb{R}^+ \times \mathbb{V}_\delta$ such that

$$a(w_\delta, v_\delta; \mu) = \lambda\ (w_\delta, v_\delta)_{\mathbb{V}}, \qquad \forall v_\delta \in \mathbb{V}_\delta.$$

This can be translated to find the smallest eigenvalue of the generalised eigenvalue problem

$$\mathbf{A}_\delta^\mu \mathrm{v}_\delta = \lambda\ \mathbf{M}_\delta \mathrm{v}_\delta,$$

where $(\mathbf{A}_\delta^\mu)_{ij} = a(\varphi_j, \varphi_i; \mu)$, $(\mathbf{M}_\delta)_{ij} = (\varphi_j, \varphi_i)_{\mathbb{V}}$ and recall that $\{\varphi_i\}_{i=1}^{N_\delta}$ is a basis of $\mathbb{V}_\delta$.

Depends on N

Computing the stability constant

How to compute $\|\hat{r}_\delta(\mu)\|_{\mathbb{V}}$?

Let

$$\boxed{\mathrm{r}_\delta^\mu = \mathrm{f}_\delta^\mu - \mathbf{A}_\delta^\mu \ \mathbf{B} \ \mathrm{u}_{\mathtt{rb}}^\mu \in \mathbb{R}^{N_\delta},}$$

be the residual vector of the N_δ-dimensional linear system

$$\boxed{\mathbf{A}_\delta^\mu \ \mathrm{u}_\delta^\mu = \mathrm{f}_\delta^\mu}$$

for the truth solution for a particular μ. The quantity $\mathbf{B} \ \mathrm{u}_{\mathtt{rb}}^\mu$ is the representation of $u_{\mathtt{rb}}(\mu)$ in the basis $\{\varphi_i\}_{i=1}^{N_\delta}$.

Then

$$\boxed{\|\hat{r}_\delta(\mu)\|_{\mathbb{V}} = \sqrt{(\mathrm{r}_\delta^\mu)^T \ \mathbf{M}_\delta^{-1} \ \mathrm{r}_\delta^\mu}.}$$

The quantity $\|\hat{r}_\delta(\mu)\|_{\mathbb{V}}$ can thus be computed but the costs depends on N_δ.

Computing the stability constant

How to compute $\|\hat{r}_\delta(\mu)\|_{\mathbb{V}}$?

Let

$$\boxed{\mathrm{r}_\delta^\mu = \mathrm{f}_\delta^\mu - \mathbf{A}_\delta^\mu \, \mathbf{B} \, \mathrm{u}_{\mathtt{rb}}^\mu \in \mathbb{R}^{N_\delta},}$$

be the residual vector of the N_δ-dimensional linear system

$$\boxed{\mathbf{A}_\delta^\mu \, \mathrm{u}_\delta^\mu = \mathrm{f}_\delta^\mu}$$

for the truth solution for a particular μ. The quantity $\mathbf{B} \, \mathrm{u}_{\mathtt{rb}}^\mu$ is the representation of $u_{\mathtt{rb}}(\mu)$ in the basis $\{\varphi_i\}_{i=1}^{N_\delta}$.

Then

$$\boxed{\|\hat{r}_\delta(\mu)\|_{\mathbb{V}} = \sqrt{(\mathrm{r}_\delta^\mu)^T \, \mathbf{M}_\delta^{-1} \, \mathrm{r}_\delta^\mu}.}$$

The quantity $\|\hat{r}_\delta(\mu)\|_{\mathbb{V}}$ can thus be computed but the costs depends on N_δ.

Strategy: Use the affine decomposition in combination with pre-computations that are μ-independent.

Computing the stability constant

Let us thus define

$$\eta(\mu) = \frac{\|\hat{r}_\delta(\mu)\|_{\mathbb{V}}}{\alpha_{\mathtt{LB}}(\mu)}$$

Theorem: We just proved that there holds

$$\|u_\delta(\mu) - u_{\mathtt{rb}}(\mu)\|_{\mathbb{V}} \leq \eta(\mu).$$

Theorem: The estimator is efficient

$$\eta(\mu) \leq \frac{\gamma_\delta(\mu)}{\alpha_{\mathtt{LB}}(\mu)} \|u_\delta(\mu) - u_{\mathtt{rb}}(\mu)\|_{\mathbb{V}}.$$

Computing the stability constant

Let us thus define

$$\eta(\mu) = \frac{\|\hat{r}_\delta(\mu)\|_{\mathbb{V}}}{\alpha_{\mathtt{LB}}(\mu)}$$

Theorem: We just proved that there holds

$$\|u_\delta(\mu) - u_{\mathtt{rb}}(\mu)\|_{\mathbb{V}} \leq \eta(\mu).$$

Theorem: The estimator is efficient

$$\eta(\mu) \leq \frac{\gamma_\delta(\mu)}{\alpha_{\mathtt{LB}}(\mu)} \|u_\delta(\mu) - u_{\mathtt{rb}}(\mu)\|_{\mathbb{V}}.$$

Proposition. In the compliant case, i.e. $\ell(\cdot;\mu) = f(\cdot;\mu)$ and a symmetric, there holds that

$$s_\delta(\mu) - s_{\mathtt{rb}}(\mu) = \|u_\delta(\mu) - u_{\mathtt{rb}}(\mu)\|_\mu^2,$$

for all $\mu \in \mathbb{P}$.

Computing the stability constant

Recall:

$$\alpha_\delta(\mu) := \inf_{v_\delta \in \mathbb{V}_\delta} \frac{a(v_\delta, v_\delta; \mu)}{\|v_\delta\|_{\mathbb{V}}^2}.$$

Goal: Design an off-line/on-line procedure where the on-line part consists of

$$\mu \to \alpha_{\mathrm{LB}}(\mu)$$

such that $0 < \alpha_{\mathrm{LB}}(\mu) \leq \alpha_\delta(\mu)$ and in a fashion that is independent of N_δ.

$\alpha_\delta(\mu)$ is the smallest eigenvalue of: find $(\lambda, w_\delta) \in \mathbb{R}^+ \times \mathbb{V}_\delta$ such that

$$a(w_\delta, v_\delta; \mu) = \lambda\, (w_\delta, v_\delta)_{\mathbb{V}}, \qquad \forall v_\delta \in \mathbb{V}_\delta.$$

which can be translated to find the smallest eigenvalue of the generalised eigenvalue problem

$$\mathbf{A}_\delta^\mu \mathrm{v}_\delta = \lambda\, \mathbf{M}_\delta \mathrm{v}_\delta,$$

where $(\mathbf{A}_\delta^\mu)_{ij} = a(\varphi_j, \varphi_i; \mu)$, $(\mathbf{M}_\delta)_{ij} = (\varphi_j, \varphi_i)_{\mathbb{V}}$ and recall that $\{\varphi_i\}_{i=1}^{N_\delta}$ is a basis of $\mathbb{V}_\delta$.

Computing the stability constant

Special case: If the decomposition

$$a(w, v; \mu) = \sum_{q=1}^{Q_{\mathtt{a}}} \theta_{\mathtt{a}}^q(\mu)\, a_q(w, v)$$

is such that

$$\begin{aligned} \theta_{\mathtt{a}}^q(\mu) &> 0, && \forall \mu \in \mathbb{P}, \quad q = 1, \dots, Q_{\mathtt{a}}, \\ a_q(v_\delta, v_\delta) &\geq 0, && \forall v_\delta \in \mathbb{V}_\delta, \quad q = 1, \dots, Q_{\mathtt{a}}. \end{aligned}$$

we call the bilinear form $a(\cdot, \cdot; \mu)$ to be **parametrically coercive**.

Example: Heat conduction on thermal blocks

$$a(w, v; \mu) = \sum_{q=1}^{15} \mu_q \int_{\mathcal{R}_q} \nabla w \cdot \nabla v + \int_{\mathcal{R}_{16}} \nabla w \cdot \nabla v,$$

$$a_q(w, v; \mu) = \int_{\mathcal{R}_q} \nabla w \cdot \nabla v, \qquad \theta_{\mathtt{a}}^q(\mu) = \mu_q, \qquad q = 1, \dots, 15,$$

$$a_{16}(w, v; \mu) = \int_{\mathcal{R}_{16}} \nabla w \cdot \nabla v, \qquad \theta_{\mathtt{a}}^{16}(\mu) = 1,$$

Computing the stability constant

Develop

$$
\begin{aligned}
\alpha_\delta(\mu) &= \inf_{v_\delta \in \mathbb{V}_\delta} \frac{a(v_\delta, v_\delta; \mu)}{\|v_\delta\|_{\mathbb{V}}^2} \\
&= \inf_{v_\delta \in \mathbb{V}_\delta} \sum_{q=1}^{Q_\mathsf{a}} \theta_\mathsf{a}^q(\mu) \frac{a_q(v_\delta, v_\delta)}{\|v_\delta\|_{\mathbb{V}}^2} \\
&= \inf_{v_\delta \in \mathbb{V}_\delta} \sum_{q=1}^{Q_\mathsf{a}} \frac{\theta_\mathsf{a}^q(\mu)}{\theta_\mathsf{a}^q(\mu')} \theta_\mathsf{a}^m(\mu') \frac{a_q(v_\delta, v_\delta)}{\|v_\delta\|_{\mathbb{V}}^2} \\
&\geq \inf_{v_\delta \in \mathbb{V}_\delta} \min_{q=1,\dots,Q_\mathsf{a}} \frac{\theta_\mathsf{a}^q(\mu)}{\theta_\mathsf{a}^q(\mu')} \sum_{q=1}^{Q_\mathsf{a}} \theta_\mathsf{a}^q(\mu') \frac{a_q(v_\delta, v_\delta)}{\|v_\delta\|^2} \\
&= \min_{q=1,\dots,Q_\mathsf{a}} \frac{\theta_\mathsf{a}^q(\mu)}{\theta_\mathsf{a}^q(\mu')} \inf_{v_\delta \in \mathbb{V}_\delta} \sum_{q=1}^{Q_\mathsf{a}} \theta_\mathsf{a}^q(\mu') \frac{a_q(v_\delta, v_\delta)}{\|v_\delta\|_{\mathbb{V}}^2} \\
&= \alpha_\delta(\mu') \min_{q=1,\dots,Q_\mathsf{a}} \frac{\theta_\mathsf{a}^q(\mu)}{\theta_\mathsf{a}^q(\mu')} := \alpha_{\mathrm{LB}}(\mu).
\end{aligned}
$$

Then

$$
\alpha_{\mathrm{LB}}(\mu) := \underbrace{\alpha_\delta(\mu') \min_{q=1,\dots,Q_\mathsf{a}} \frac{\theta_\mathsf{a}^q(\mu)}{\theta_\mathsf{a}^q(\mu')}}_{\in(0,\infty)} \leq \alpha_\delta(\mu).
$$

Computing the stability constant

Min-Θ approach

$$\alpha_{\mathrm{LB}}(\mu) := \underbrace{\alpha_\delta(\mu') \min_{q=1,\dots,Q_{\mathtt{a}}} \frac{\theta_{\mathtt{a}}^q(\mu)}{\theta_{\mathtt{a}}^q(\mu')}}_{\in(0,\infty)} \leq \alpha_\delta(\mu).$$

Computing the stability constant

Min-Θ approach

$$\alpha_{\mathrm{LB}}(\mu) := \underbrace{\alpha_\delta(\mu') \min_{q=1,\ldots,Q_{\mathtt{a}}} \frac{\theta^q_{\mathtt{a}}(\mu)}{\theta^q_{\mathtt{a}}(\mu')}}_{\in(0,\infty)} \leq \alpha_\delta(\mu).$$

Multiple anchor points

$$\alpha_{\mathrm{LB}}(\mu) = \max_{k=1,\ldots,K} \left(\alpha_\delta(\mu_k) \min_{q=1,\ldots,Q_{\mathtt{a}}} \frac{\theta^q_{\mathtt{a}}(\mu)}{\theta^q_{\mathtt{a}}(\mu_k)} \right).$$

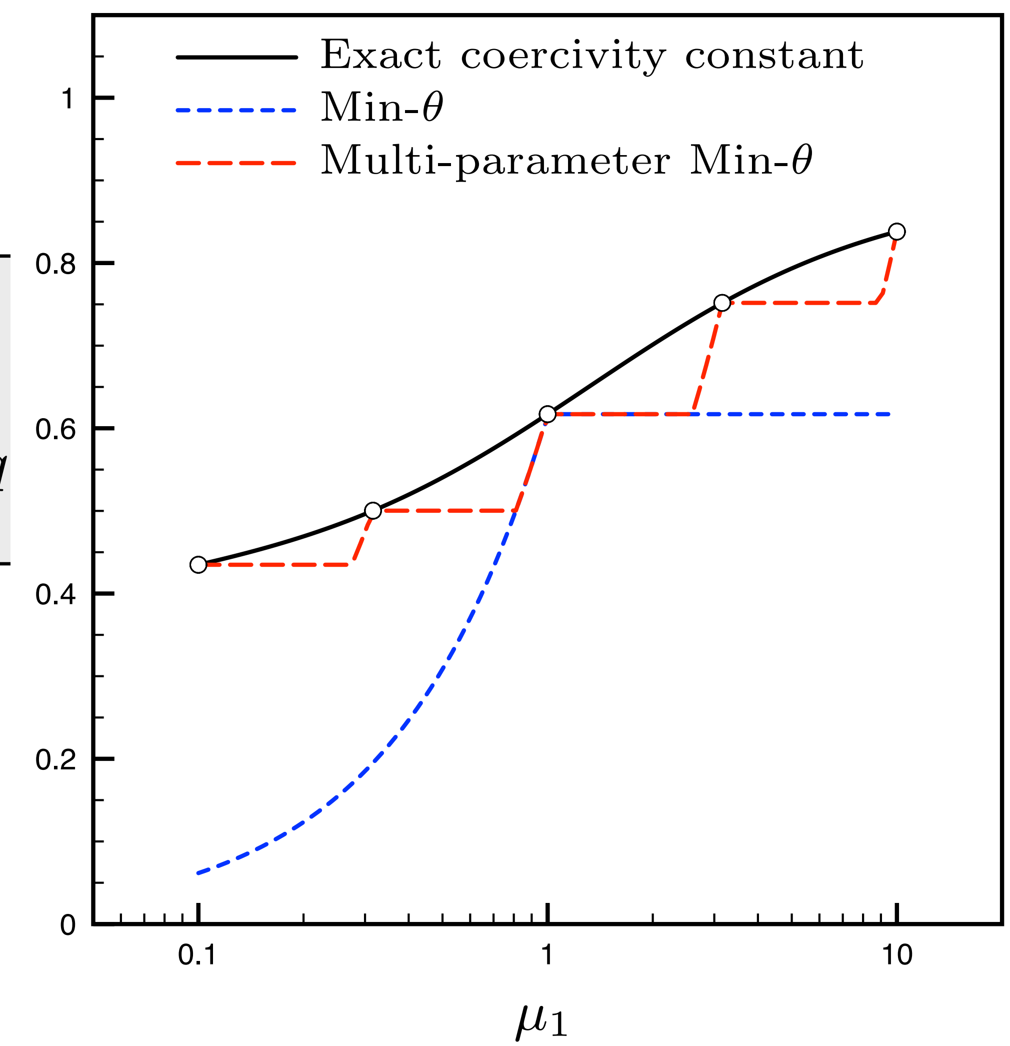
Computing the stability constant

Min-Θ approach

$$\alpha_{\mathrm{LB}}(\mu) := \underbrace{\alpha_\delta(\mu') \min_{q=1,\dots,Q_{\mathtt{a}}} \frac{\theta_{\mathtt{a}}^q(\mu)}{\theta_{\mathtt{a}}^q(\mu')}}_{\in(0,\infty)} \leq \alpha_\delta(\mu).$$

Multiple anchor points

$$\alpha_{\mathrm{LB}}(\mu) = \max_{k=1,\dots,K} \Big(\alpha_\delta(\mu_k) \quad {}_q$$



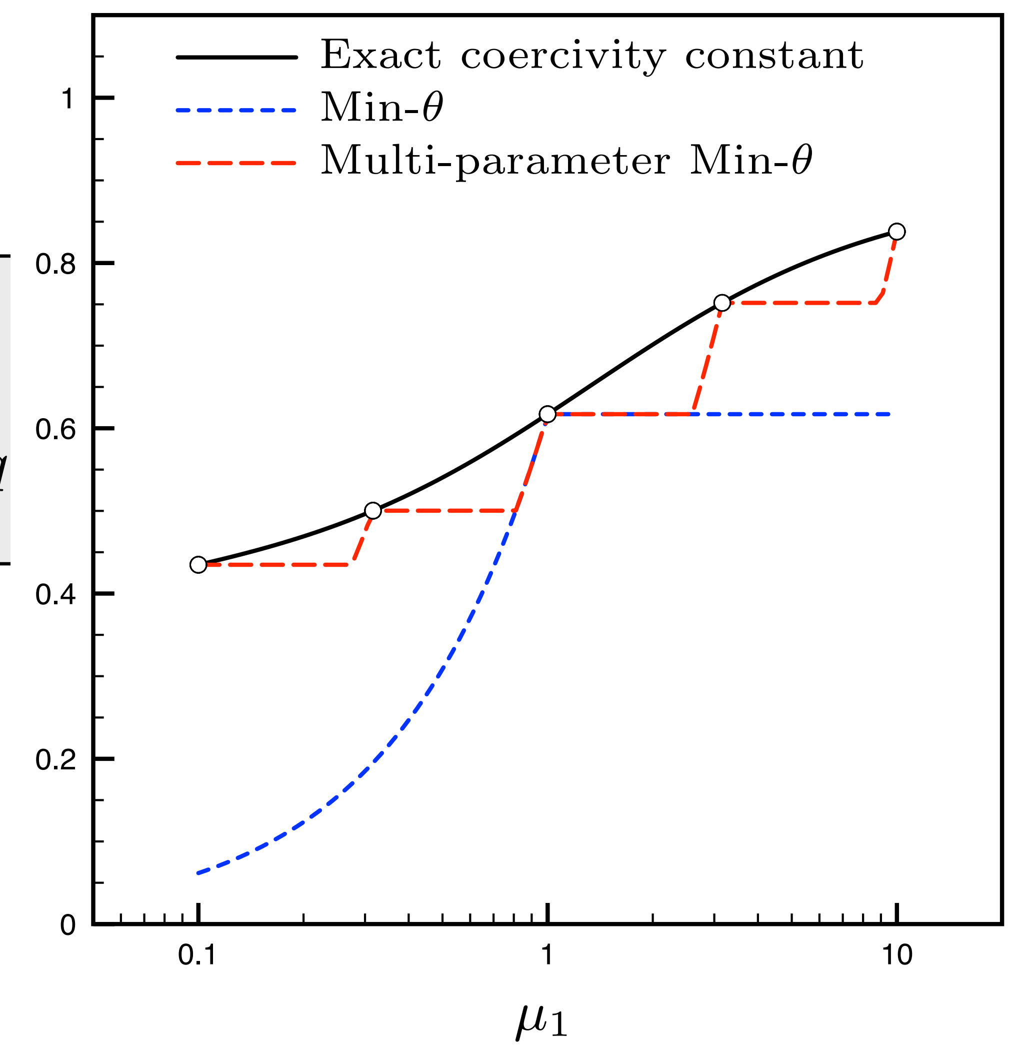
Computing the stability constant

Min-Θ approach

$$\alpha_{\mathrm{LB}}(\mu) := \underbrace{\alpha_\delta(\mu') \min_{q=1,\dots,Q_{\mathtt{a}}} \frac{\theta_{\mathtt{a}}^q(\mu)}{\theta_{\mathtt{a}}^q(\mu')}}_{\in(0,\infty)} \leq \alpha_\delta(\mu).$$

Multiple anchor points

$$\alpha_{\mathrm{LB}}(\mu) = \max_{k=1,\dots,K} \Big(\alpha_\delta(\mu_k) \quad {}_{q}$$



Limited and not very accurate

Computing the stability constant (SCM)

Consider the coercive case (for simplicity)

$$\alpha(\mu) = \inf_{v_h \in X_h} \frac{a_h(v_h, \mu, v_h)}{\|v_h\|^2} = \inf_{v_h \in X_h} \sum_{k=1}^{Q_a} \Theta_k^a(\mu) \frac{a_k(v_h, v_h)}{\|v_h\|^2}$$

and define

$$\mathcal{Y} = \left\{ y \in \mathsf{R}^{Q_a} | \exists w_h \in X_h ; y_k = \frac{a_k(w_h, w_h)}{\|w_h\|^2} \right\}$$

Computing the stability constant (SCM)

Consider the coercive case (for simplicity)

$$\alpha(\mu) = \inf_{v_h \in X_h} \frac{a_h(v_h, \mu, v_h)}{\|v_h\|^2} = \inf_{v_h \in X_h} \sum_{k=1}^{Q_a} \Theta_k^a(\mu) \frac{a_k(v_h, v_h)}{\|v_h\|^2}$$

and define

$$\mathcal{Y} = \left\{ y \in \mathsf{R}^{Q_a} | \exists w_h \in X_h ; y_k = \frac{a_k(w_h, w_h)}{\|w_h\|^2} \right\}$$

If we have

$$\mathcal{J} : \mathsf{R}^{Q_a} \times \mathcal{D} \to \mathsf{R} \qquad \mathcal{J}(y, \mu) = \sum_{k=1}^{Q_a} Q_k^a(\mu) y_k$$

then

$$\alpha(\mu) = \min_{y \in \mathcal{Y}} \mathcal{J}(y, \mu)$$

Computing the stability constant (SCM)

Strategy: Find upper and lower bounds such that

$$\mathcal{Y}_{UB} \subset \mathcal{Y} \subset \mathcal{Y}_{LB}$$

and define

$$\alpha_{LB}(\boldsymbol{\mu}) = \min_{\boldsymbol{y} \in \mathcal{Y}_{LB}} \mathcal{I}(\boldsymbol{\mu}; \boldsymbol{u}) \qquad \alpha_{UB}(\boldsymbol{\mu}) = \min_{\boldsymbol{y} \in \mathcal{Y}_{UB}} \mathcal{I}(\boldsymbol{\mu}; \boldsymbol{u}),$$

Computing the stability constant (SCM)

Strategy: Find upper and lower bounds such that

$$\mathcal{Y}_{UB} \subset \mathcal{Y} \subset \mathcal{Y}_{LB}$$

and define

$$\alpha_{LB}(\boldsymbol{\mu}) = \min_{\boldsymbol{y} \in \mathcal{Y}_{LB}} \mathcal{I}(\boldsymbol{\mu}; \boldsymbol{u}) \qquad \alpha_{UB}(\boldsymbol{\mu}) = \min_{\boldsymbol{y} \in \mathcal{Y}_{UB}} \mathcal{I}(\boldsymbol{\mu}; \boldsymbol{u}),$$

Introduce

$$1 \leq k \leq Q_a \quad \begin{aligned} \sigma_k^- &= \inf_{v_h \in X_h} \frac{a_k(v_h, v_h)}{\|v_h\|^2} \\ \sigma_k^+ &= \sup_{v_h \in X_h} \frac{a_k(v_h, v_h)}{\|v_h\|^2} \end{aligned}$$

Computing the stability constant (SCM)

<u>Strategy</u>: Find upper and lower bounds such that

$$\mathcal{Y}_{UB} \subset \mathcal{Y} \subset \mathcal{Y}_{LB}$$

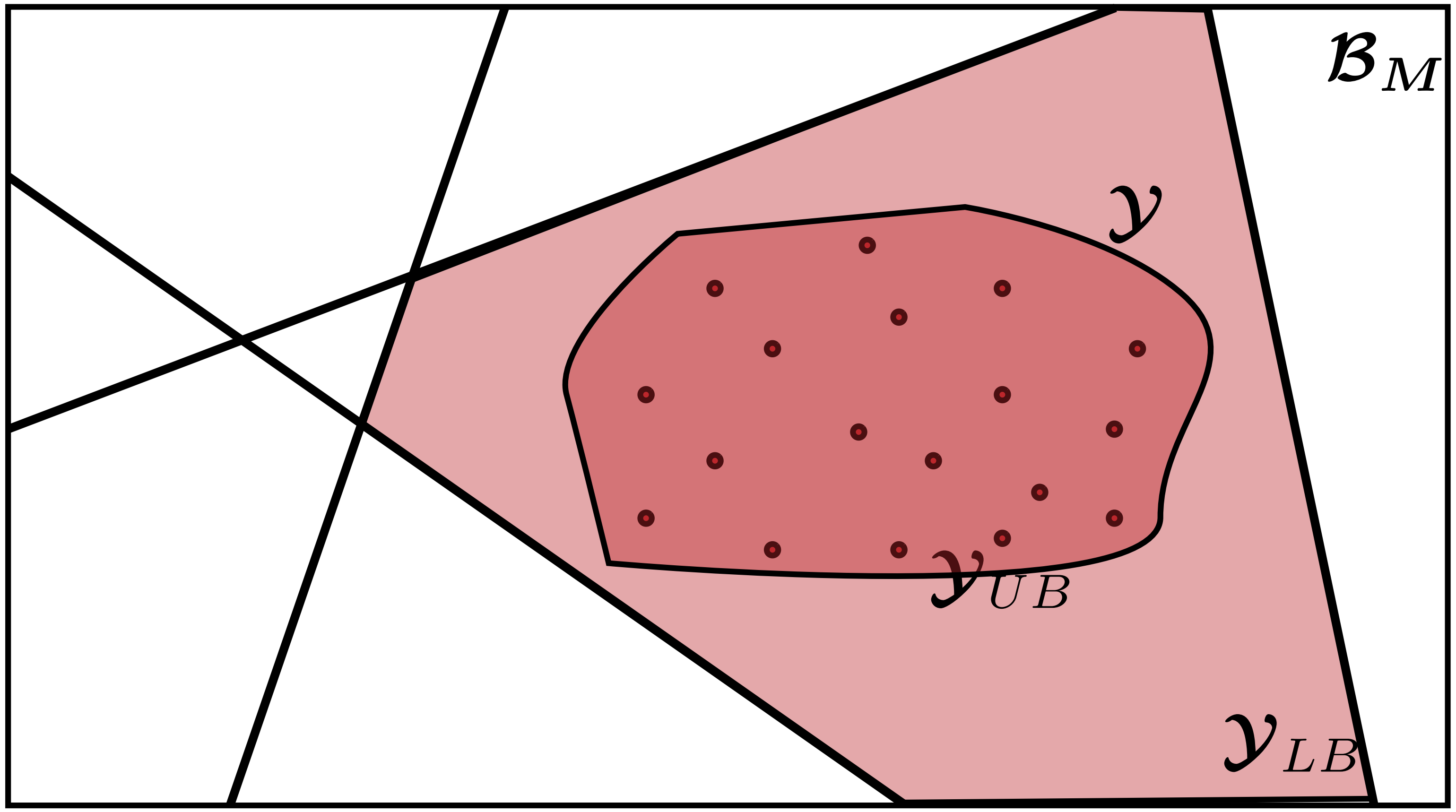
and define

$$\alpha_{LB}(\boldsymbol{\mu}) = \min_{\boldsymbol{y} \in \mathcal{Y}_{LB}} \mathcal{I}(\boldsymbol{\mu}; \boldsymbol{u}) \qquad \alpha_{UB}(\boldsymbol{\mu}) = \min_{\boldsymbol{y} \in \mathcal{Y}_{UB}} \mathcal{I}(\boldsymbol{\mu}; \boldsymbol{u}),$$

Introduce

$$1 \le k \le Q_a \qquad \sigma_k^- = \inf_{v_h \in X_h} \frac{a_k(v_h, v_h)}{\|v_h\|^2} \qquad \sigma_k^+ = \sup_{v_h \in X_h} \frac{a_k(v_h, v_h)}{\|v_h\|^2}$$

$$\mathcal{B} = \prod_{k=1}^{Q_a} [\sigma_k^-, \sigma_k^+]$$



Computing the stability constant (SCM)

Algorithm: Offline-procedure of the SCM

Input: An error tolerance `Tol`, some initial set $\mathbb{C}_1 = \{\mu_1\}$ and $n = 1$

Output: The sample points $\mathbb{C}_N = \{\mu_1, \ldots, \mu_N\}$, the corresponding coercivity constants $\alpha_\delta(\mu_n)$ and vectors y^n, $n = 1, \ldots, N$, as well as the lower bounds $\alpha_{\mathrm{LB}}^N(\mu)$ for all $\mu \in \Xi_{\mathsf{a}}$.

1. For each $\mu \in \Xi_{\mathsf{a}}$:
   - a. Compute the upper bound $\alpha_{\mathrm{UB}}^n(\mu) = \min_{\mathrm{y} \in \boldsymbol{\mathcal{Y}}_{\mathrm{UB}}^n} \mathtt{S}(\mu, \mathrm{y})$.
   - b. Compute the lower bound $\alpha_{\mathrm{LB}}^n(\mu) = \min_{\mathrm{y} \in \boldsymbol{\mathcal{Y}}_{\mathrm{LB}}^n(\mu)} \mathtt{S}(\mu, \mathrm{y})$.
   - c. Define the error estimate $\eta(\mu; \mathbb{C}_n) = 1 - \frac{\alpha_{\mathrm{LB}}^n(\mu)}{\alpha_{\mathrm{UB}}^n(\mu)}$.
2. Select $\mu_{n+1} = \operatorname{argmax}_{\mu \in \mathbb{P}} \eta(\mu; \mathbb{C}_n)$ and set $\mathbb{C}_{n+1} = \mathbb{C}_n \cup \{\mu_{n+1}\}$.
3. If $\max_{\mu \in \mathbb{P}} \eta(\mu; \mathbb{C}_n) \leq$ `Tol`, **terminate**.
4. Solve the generalized eigenvalue problem (4.26) associated with μ_{n+1}, store $\alpha_\delta(\mu_{n+1})$, y^{n+1}.
5. Set $n := n + 1$ and **goto 1.**

Online part

$$\alpha_{\mathrm{LB}}(\mu) = \min_{\mathrm{y} \in \boldsymbol{\mathcal{Y}}_{\mathrm{LB}}(\mu)} \mathtt{S}(\mu, \mathrm{y}),$$

Computing the stability constant (SCM)

The upper bound

Assumption: We know the coercivity constant $\alpha(\omega_i)$ for a given point set $C_K = \{\omega_1, \ldots, \omega_K\}$

Then define

$$\mathcal{Y}_{UB}(C_K) = \{y^*(\omega_k) | 1 \leq k \leq K\}, \quad y^*(\mu) = \arg \inf_{y \in \mathcal{Y}} \mathcal{I}(\mu, y)$$

Greedy by

$$1 - \frac{\alpha_{\mathrm{LB}}(\mu)}{\alpha_{\mathrm{UB}}(\mu)},$$

Computing the stability constant (SCM)

The upper bound

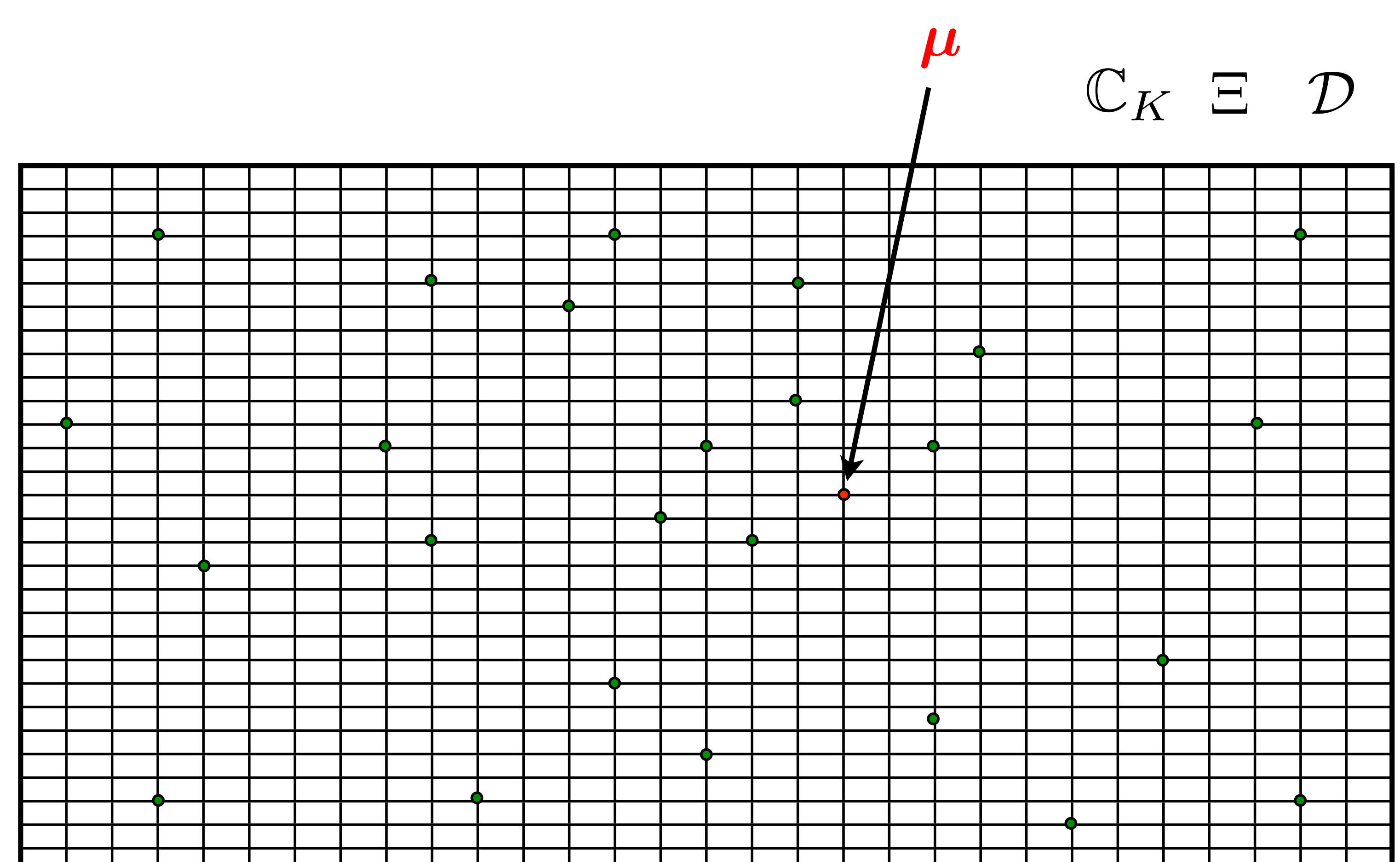
Assumption: We know the coercivity constant $\alpha(\omega_i)$ for a given point set $C_K = \{\omega_1, \ldots, \omega_K\}$

Then define

$$\mathcal{Y}_{UB}(C_K) = \{y^*(\omega_k) | 1 \le k \le K\}, \quad y^*(\mu) = \arg \inf_{y \in \mathcal{Y}} \mathcal{I}(\mu, y)$$

Greedy by

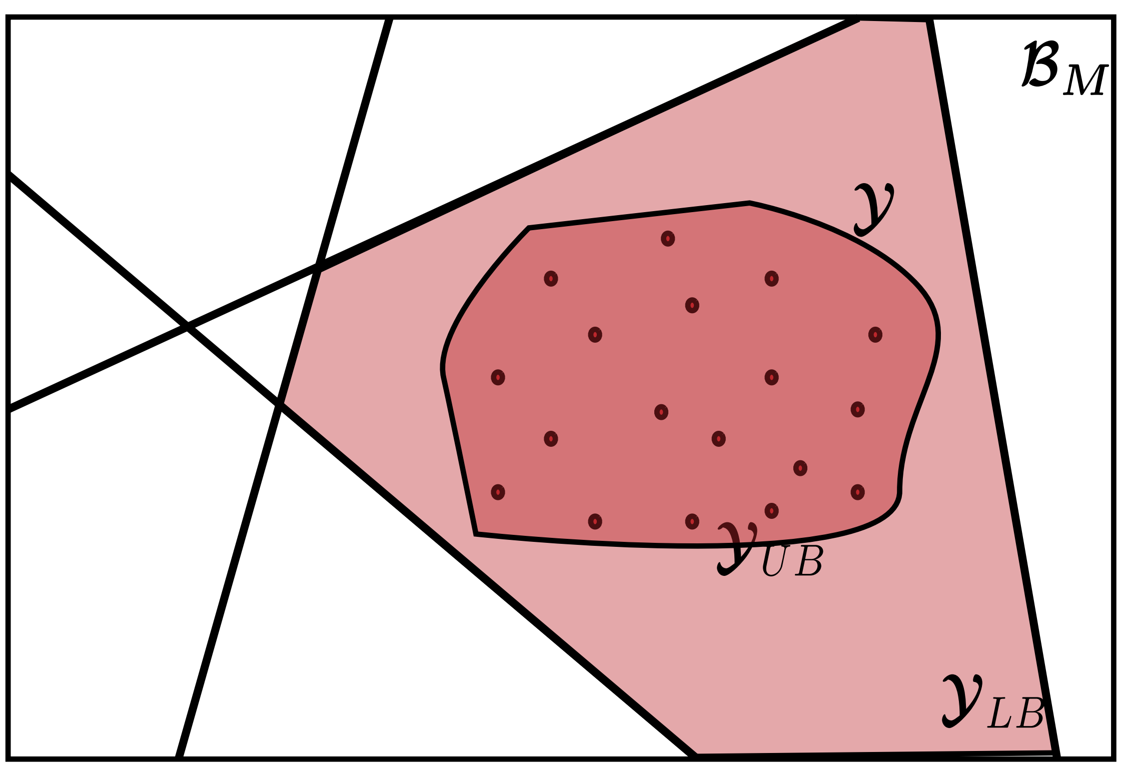
$$1 - \frac{\alpha_{\text{LB}}(\mu)}{\alpha_{\text{UB}}(\mu)},$$



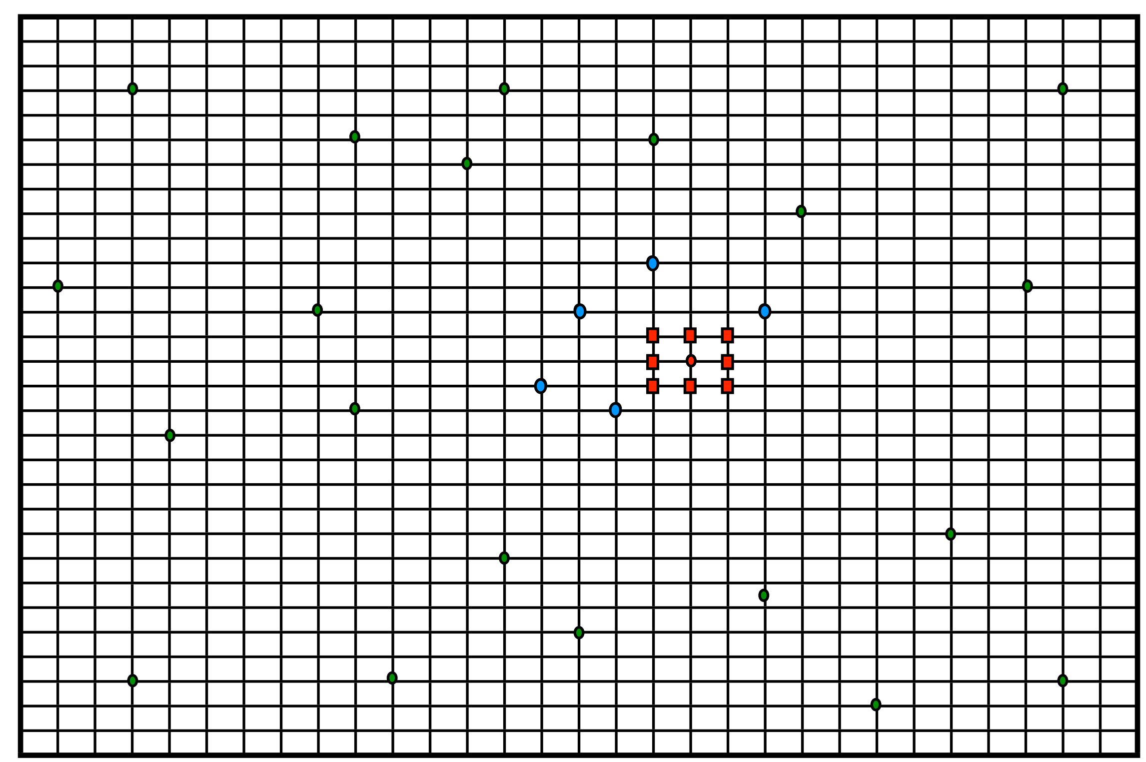
Computing the stability constant (SCM)

The lower bound is found though

$$\mathcal{Y}_{LB}(\boldsymbol{\mu};\mathbb{C}_K) = \left\{ \boldsymbol{y} \in \mathcal{B}_M \;\middle|\; \begin{array}{l} \displaystyle\sum_{m=1}^{M} \Theta^m(\boldsymbol{\mu}')\, \boldsymbol{y}_m \geq \alpha_h(\boldsymbol{\mu}'), \quad \forall \boldsymbol{\mu}' \in \mathbb{P}_{M_\alpha}(\boldsymbol{\mu};\mathbb{C}_K) \\ \displaystyle\sum_{m=1}^{M} \Theta^m(\boldsymbol{\mu}')\, \boldsymbol{y}_m \geq 0, \quad \forall \boldsymbol{\mu}' \in \mathbb{P}_{M_+}(\boldsymbol{\mu};\Xi) \end{array} \right\}$$



$\mathbb{C}_K \quad \Xi \quad \mathcal{D}$



Computing the stability constant (SCM)

The upper and lower bound computation requires

✓ Local minimization to compute upper bound

✓ Linear programming problem to compute lower bound

Computing the stability constant (SCM)

The upper and lower bound computation requires

- ✓ Local minimization to compute upper bound
- ✓ Linear programming problem to compute lower bound

Complexity is independent of $\mathcal{N}$

Computing the stability constant (SCM)

The upper and lower bound computation requires

- ✓ Local minimization to compute upper bound
- ✓ Linear programming problem to compute lower bound

Complexity is independent of $\mathcal{N}$

Set C_K is computed through greedy approach by minimizing difference between upper and lower bound estimate

$$\text{Find } \Omega_{K+1} = \arg\max_{\boldsymbol{\mu}\in\Xi} \frac{\alpha_{UB}(\boldsymbol{\mu};\mathbb{C}_K)-\alpha_{LB}(\boldsymbol{\mu};\mathbb{C}_K)}{\alpha_{UB}(\boldsymbol{\mu};\mathbb{C}_K)}.$$

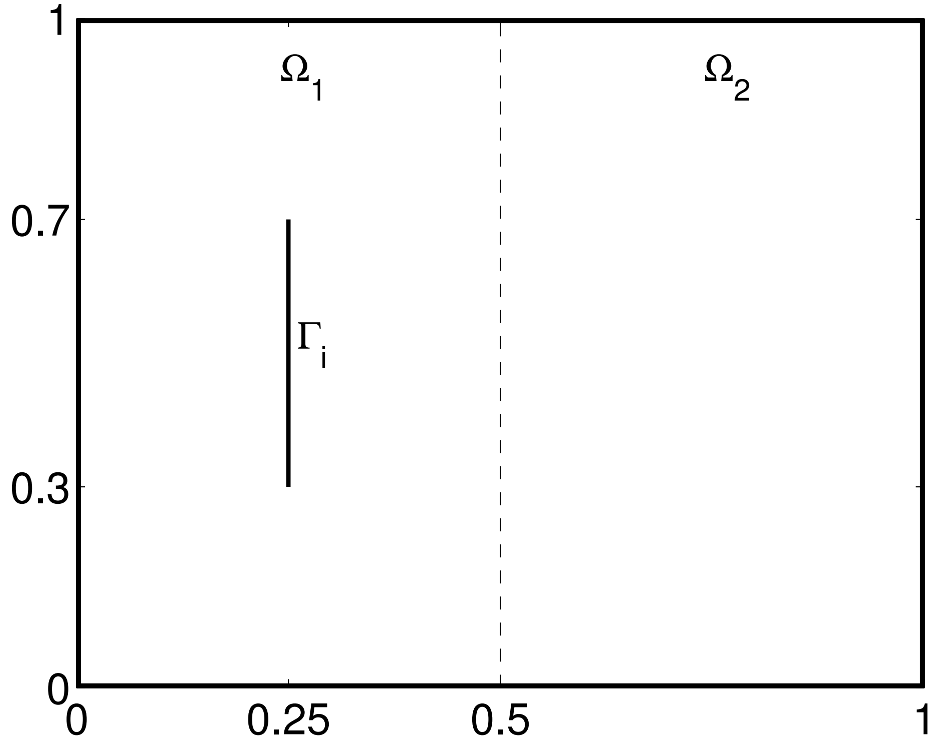
$$\text{Update } \mathbb{C}_{K+1} = \mathbb{C}_K \cup \omega_{K+1}$$

Lets see if it works

In the first examples, the lower bound on the stability does not play an essential role.

Lets see if it works

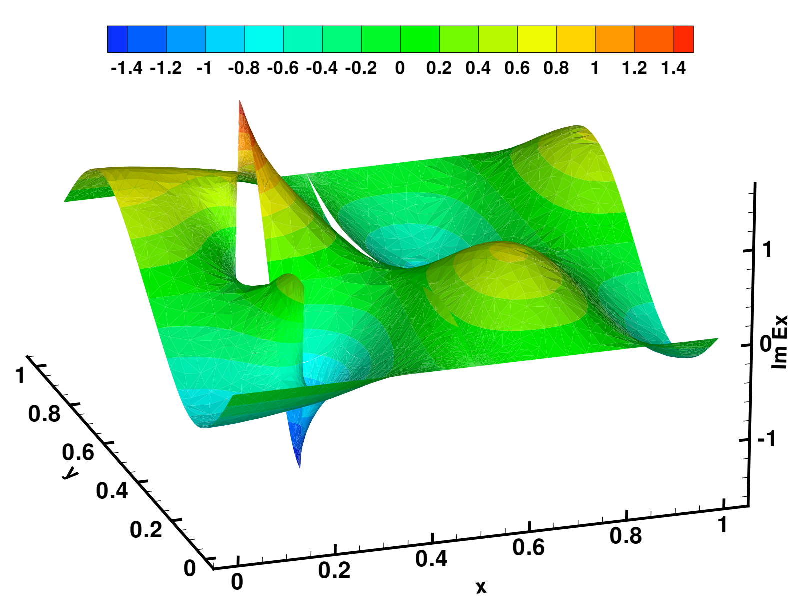
In the first examples, the lower bound on the stability does not play an essential role.

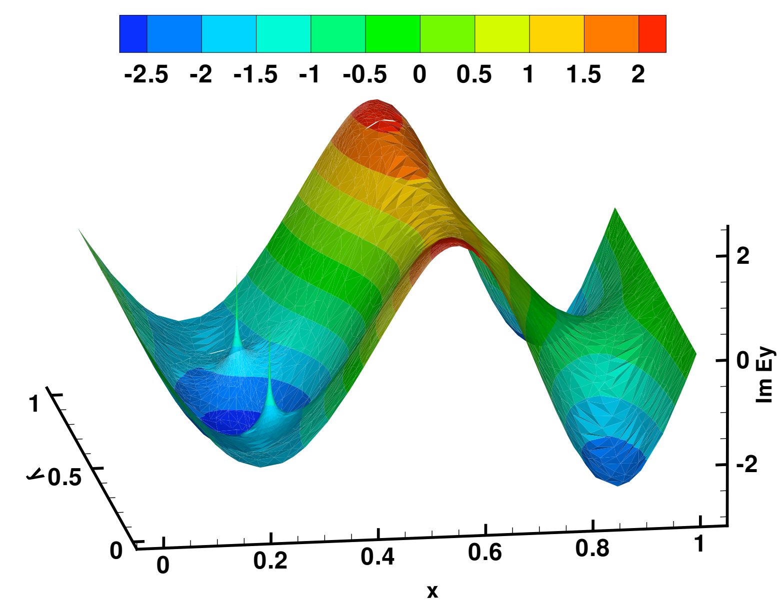


2D EM problem

Parameters are material properties in right half

$$s = \int_{\Omega_2} E_x + E_y \, dx$$



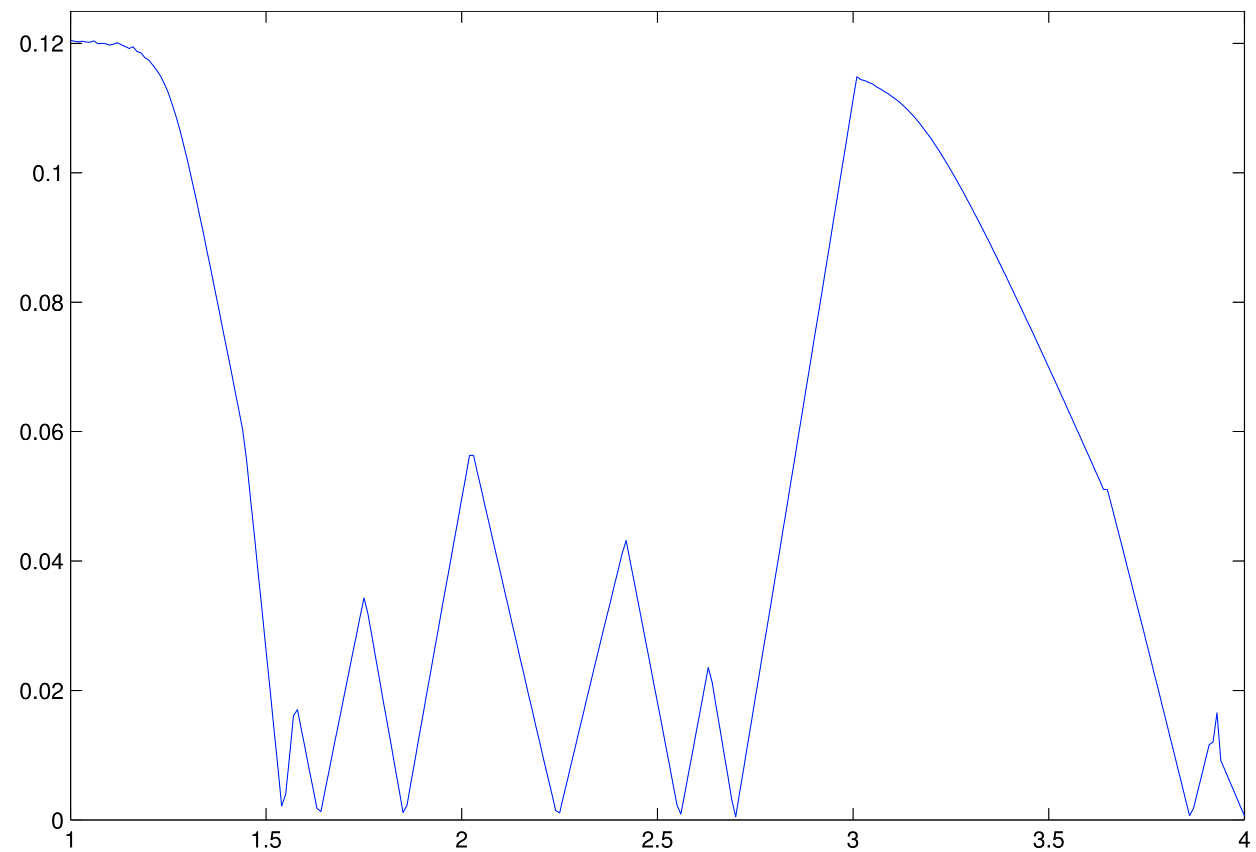


Lets see if it works

This is in fact a hard problem

$\varepsilon_2 \;:\; \omega = 5\pi/2$ is a resonance
1.5434
1.6357
1.8532
2.2456
2.5569
2.6983
3.8615
4.0033

(a) Resonances.



(b) $\beta^{pd}(\mu)$.

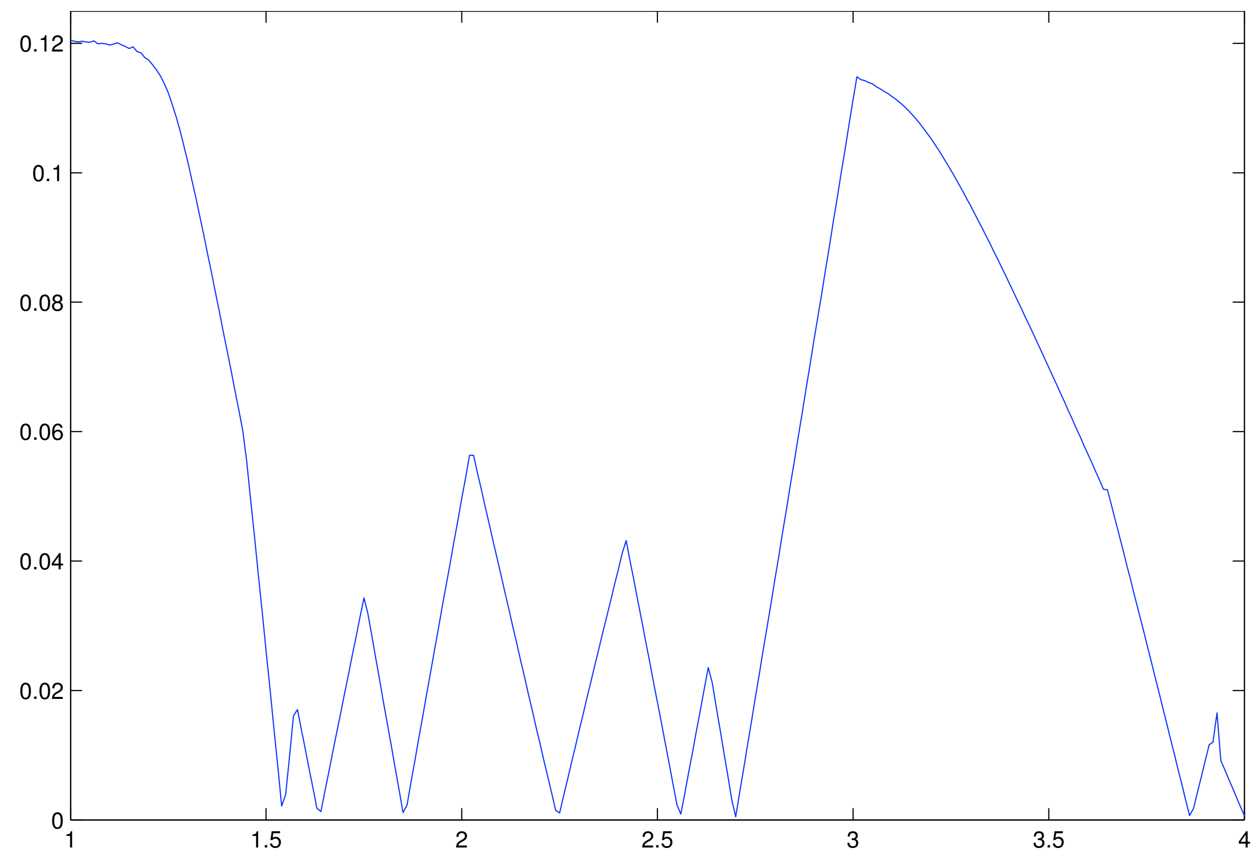
$\mathcal{N}$ = 11884, 4th order, 282 elements

Lets see if it works

This is in fact a hard problem

$\varepsilon_2 \ : \ \omega = 5\pi/2$ is a resonance
1.5434
1.6357
1.8532
2.2456
2.5569
2.6983
3.8615
4.0033

(a) Resonances.



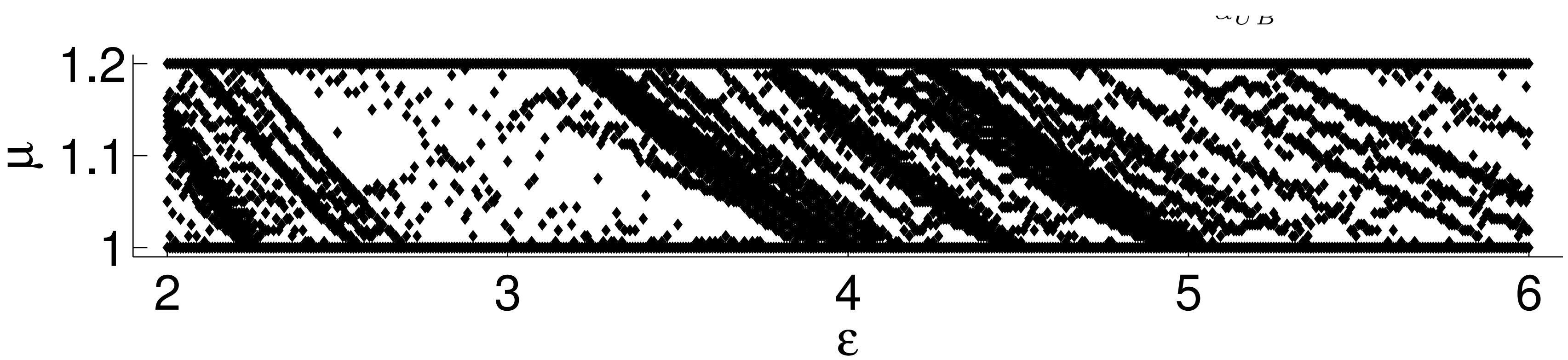
(b) $\beta^{pd}(\mu)$.

$\mathcal{N} = 11884$, 4th order, 282 elements

<u>In this case the stability constant is important</u>

Lets see if it works

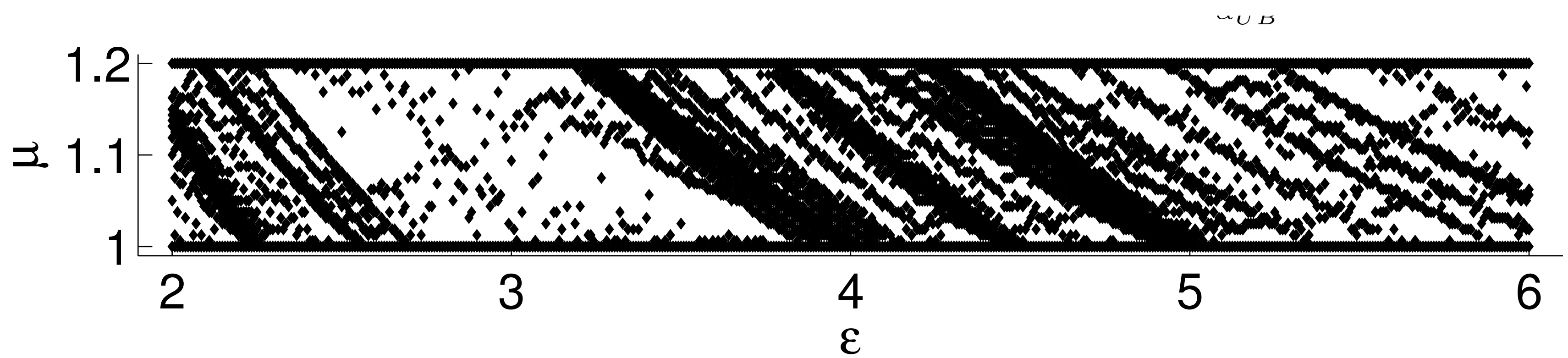
Point sets computed for lower-bound computation



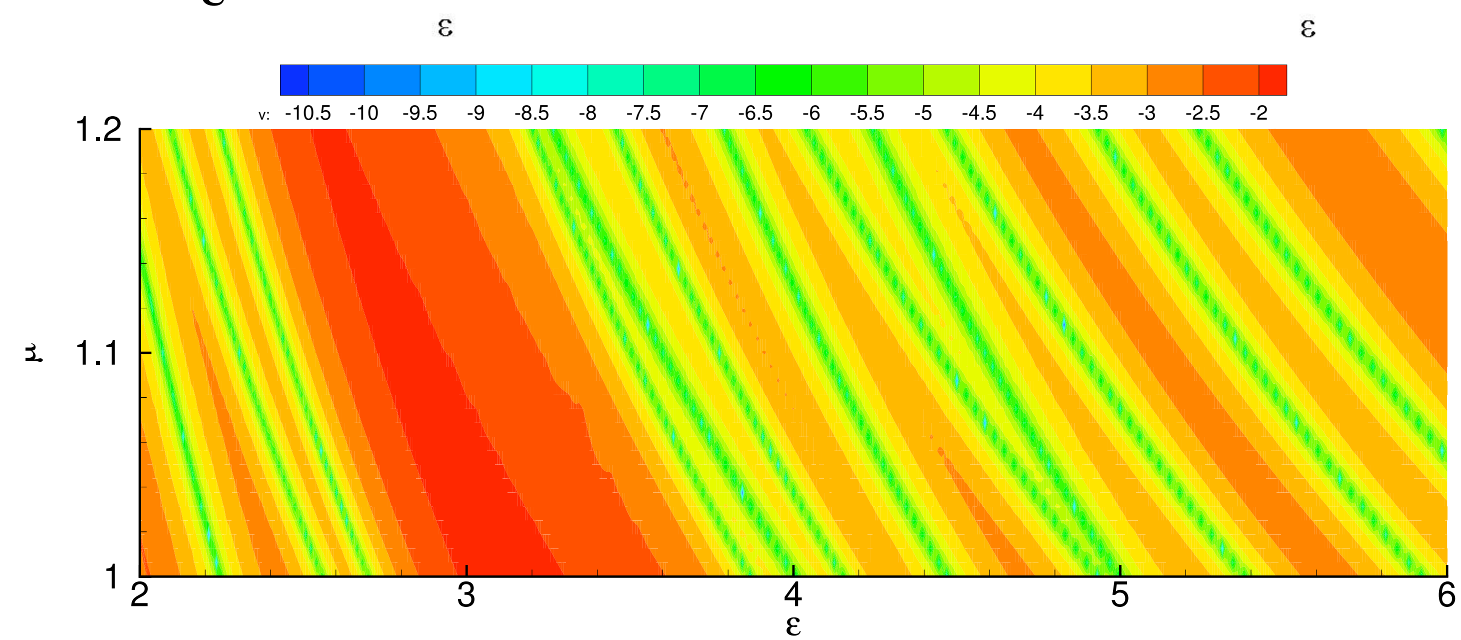
Lets see if it works

ÉCOLE POLYTECHNIQUE FÉDÉRALE DE LAUSANNE

Point sets computed for lower-bound computation

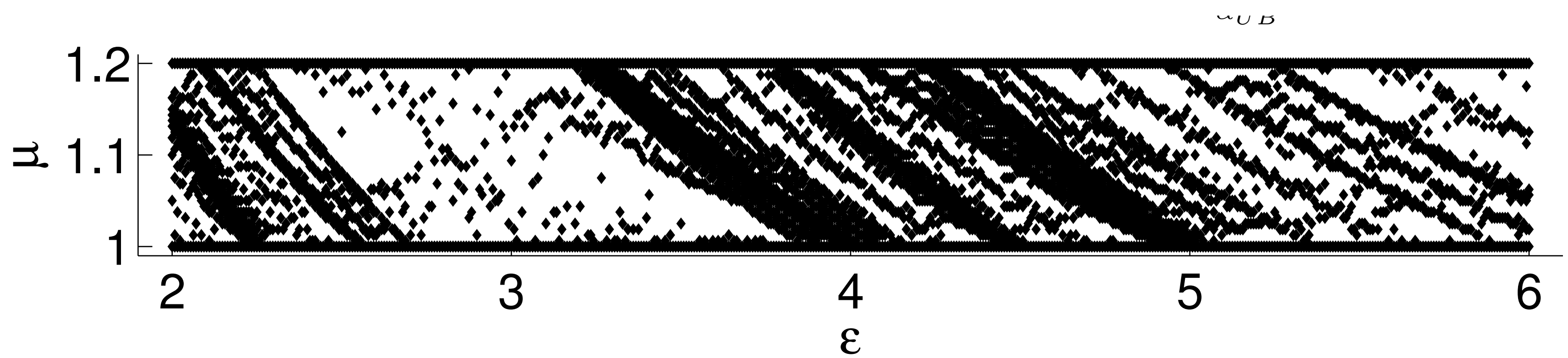


Lower bound approximation

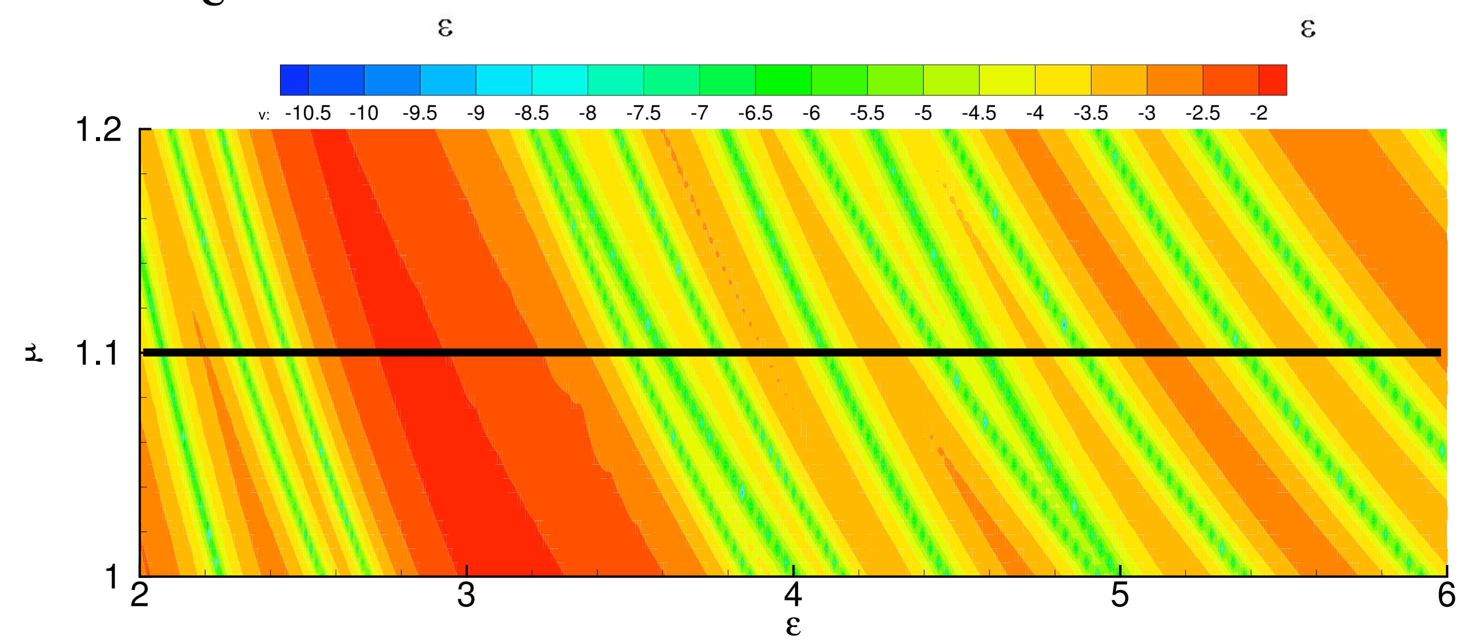


Lets see if it works

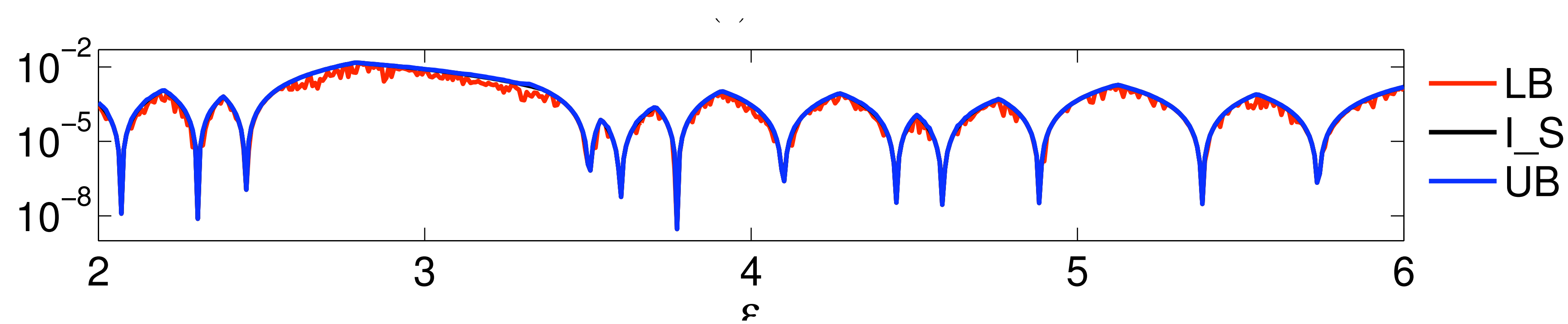
Point sets computed for lower-bound computation



Lower bound approximation

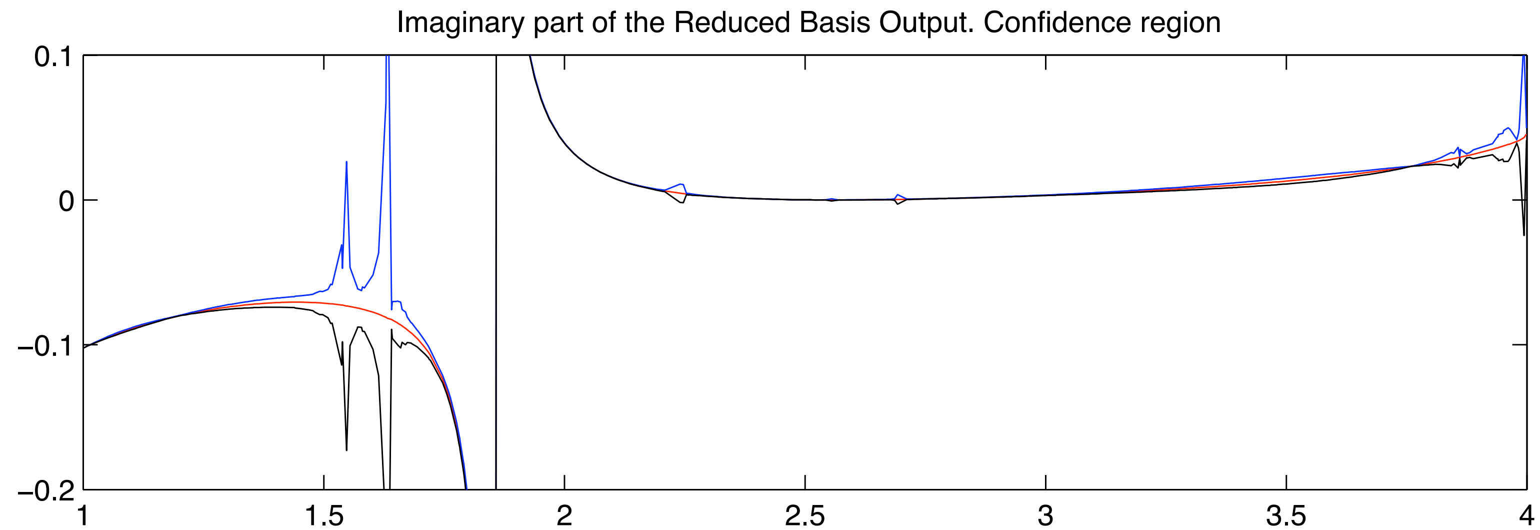


Accuracy

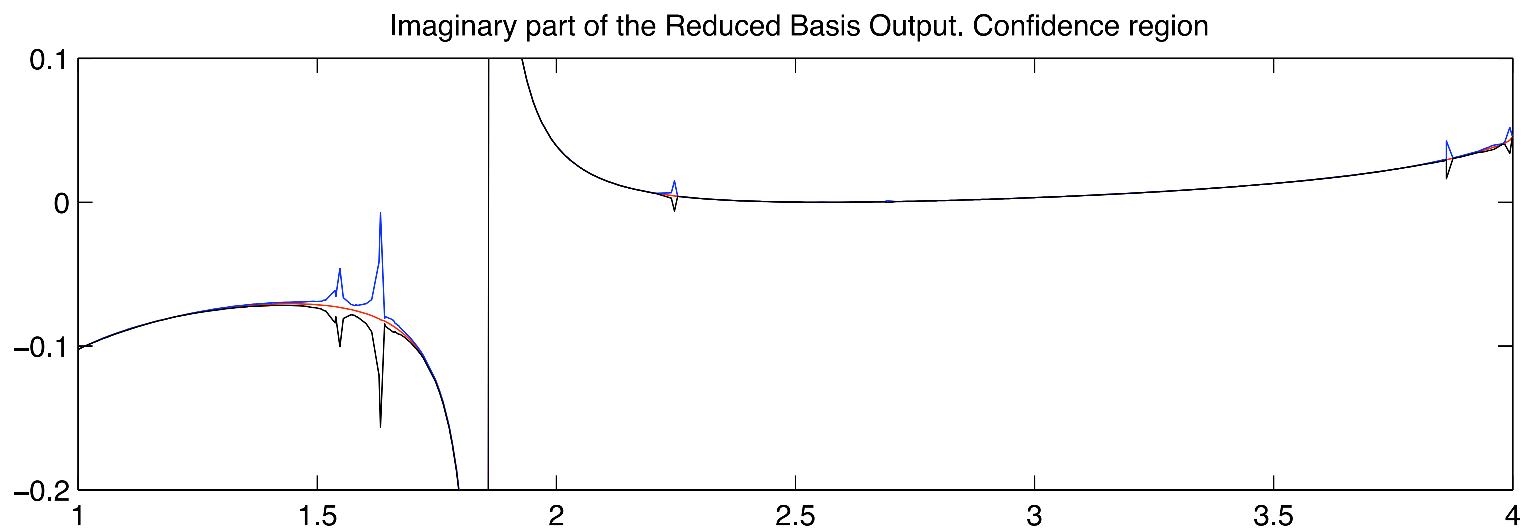


Lets see if it works

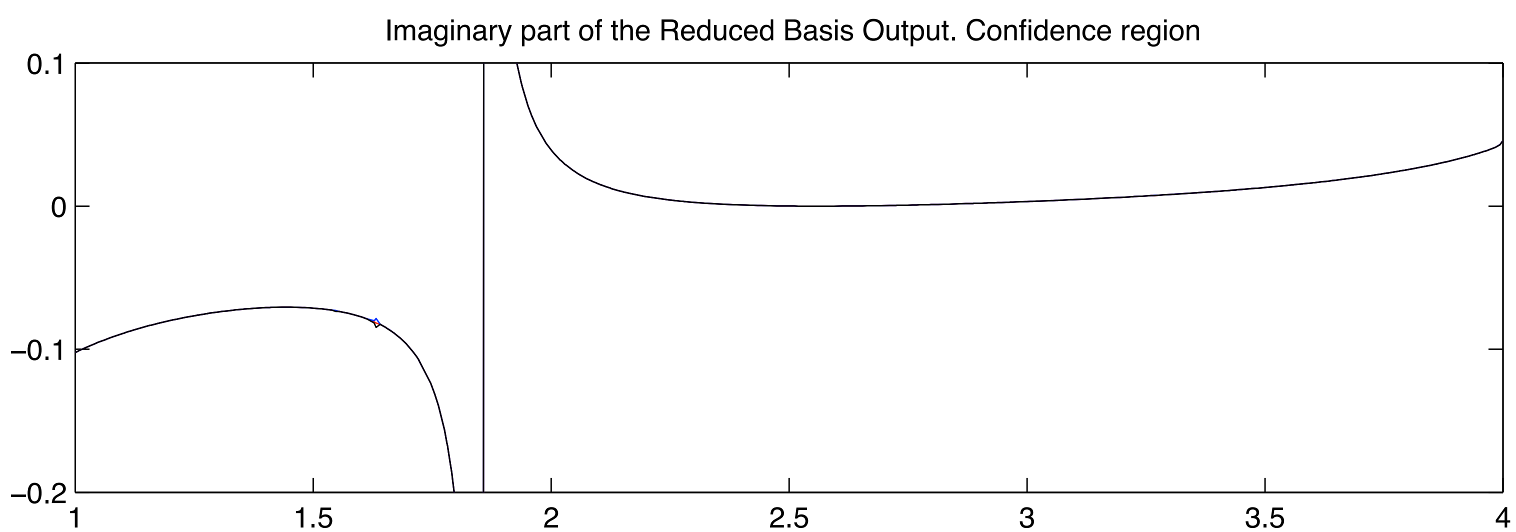
N=10



N=12

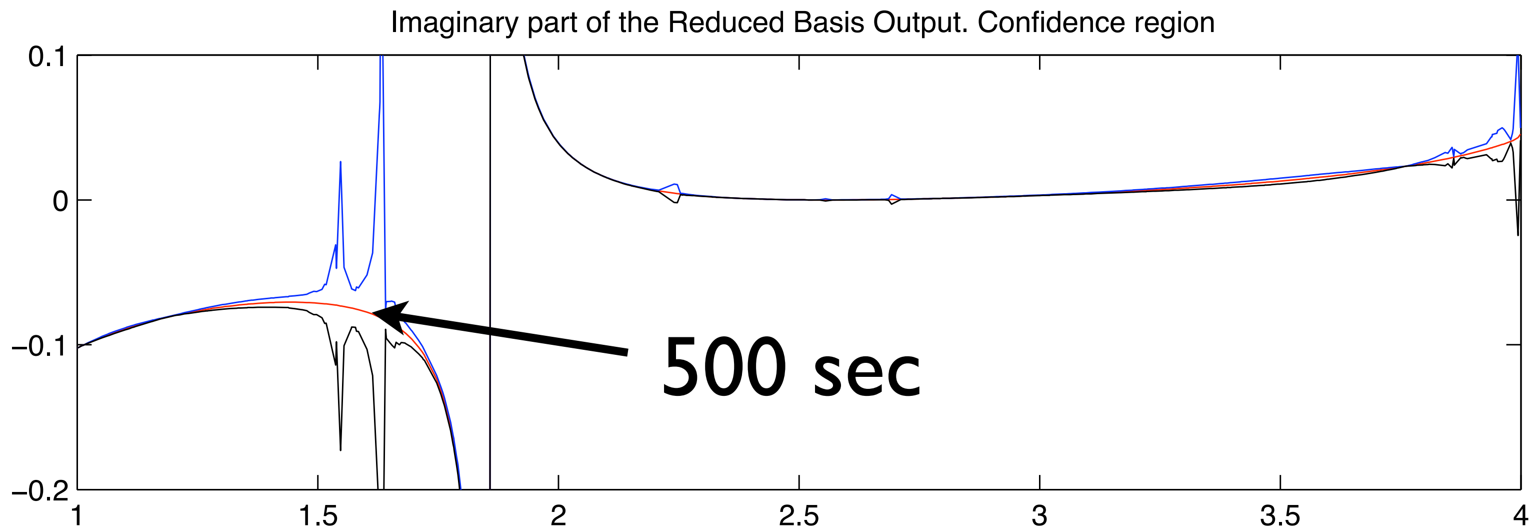


N=15

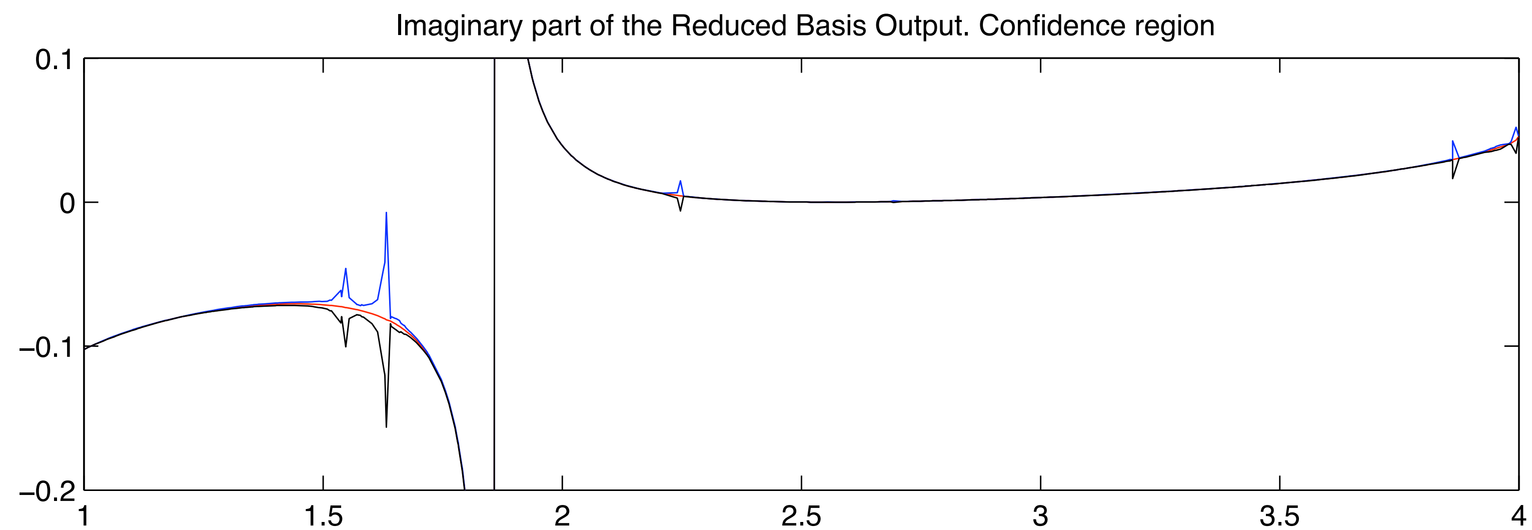


Lets see if it works

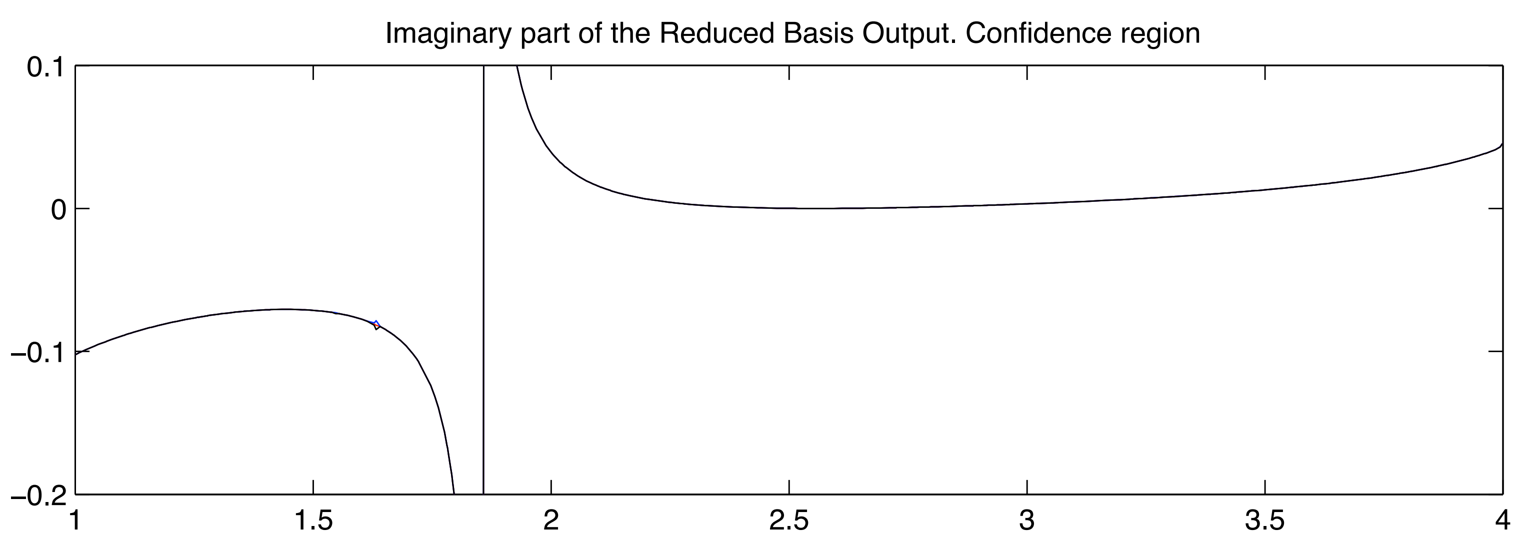
N=10



N=12

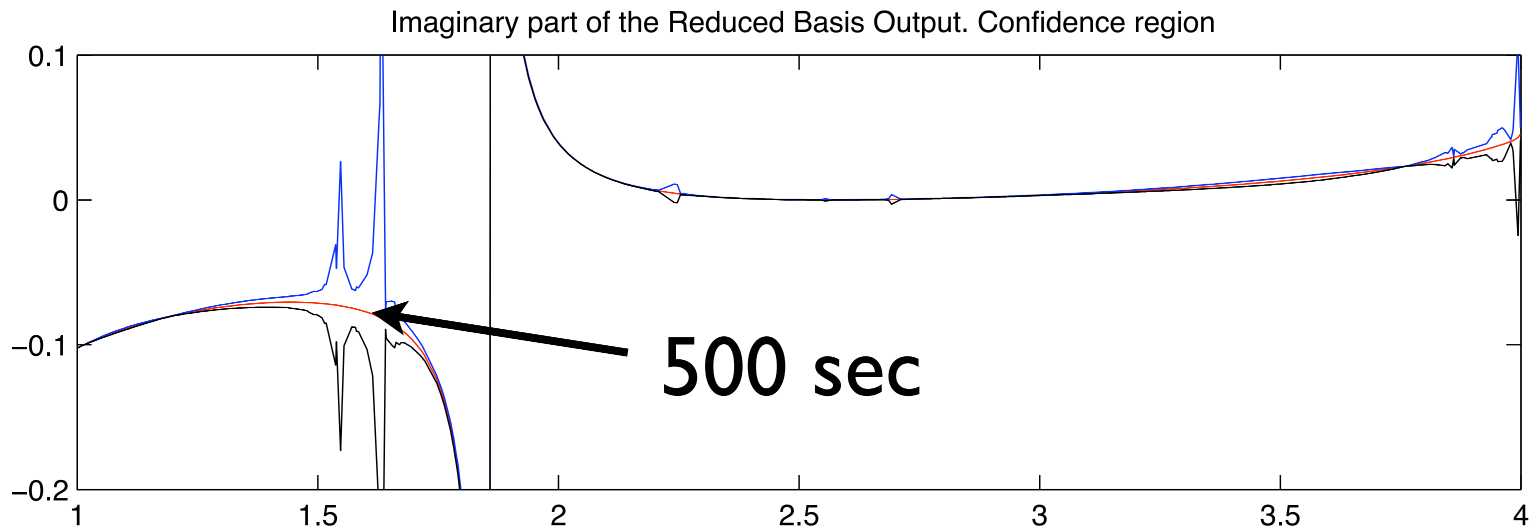


N=15



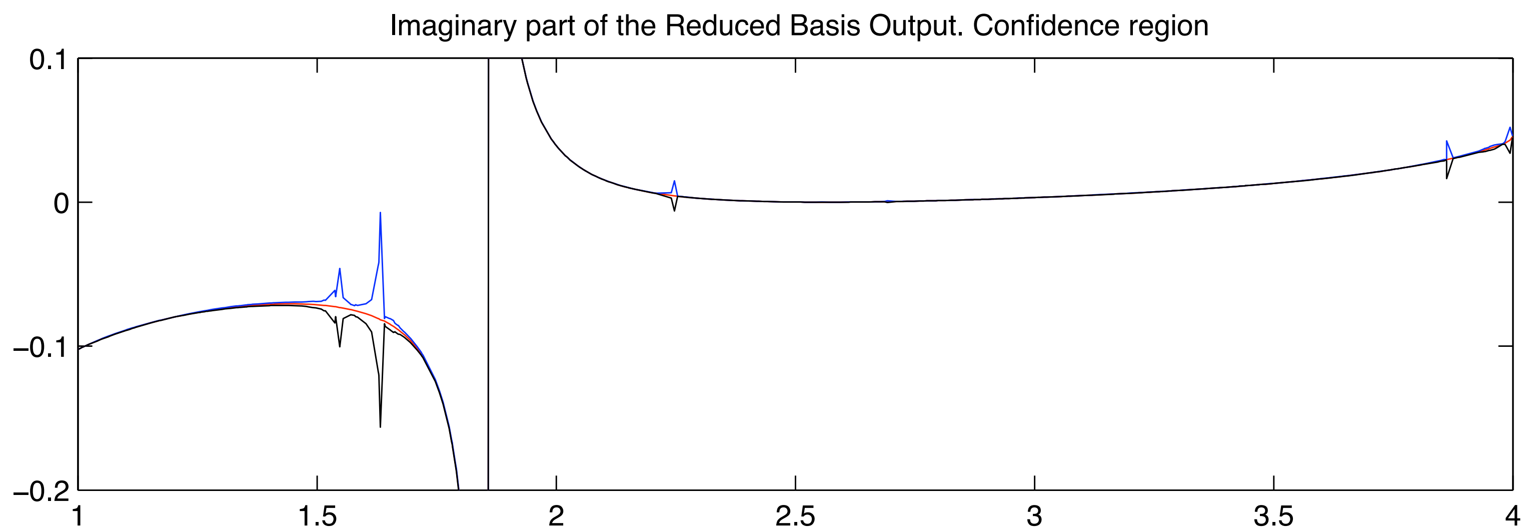
Lets see if it works

N=10

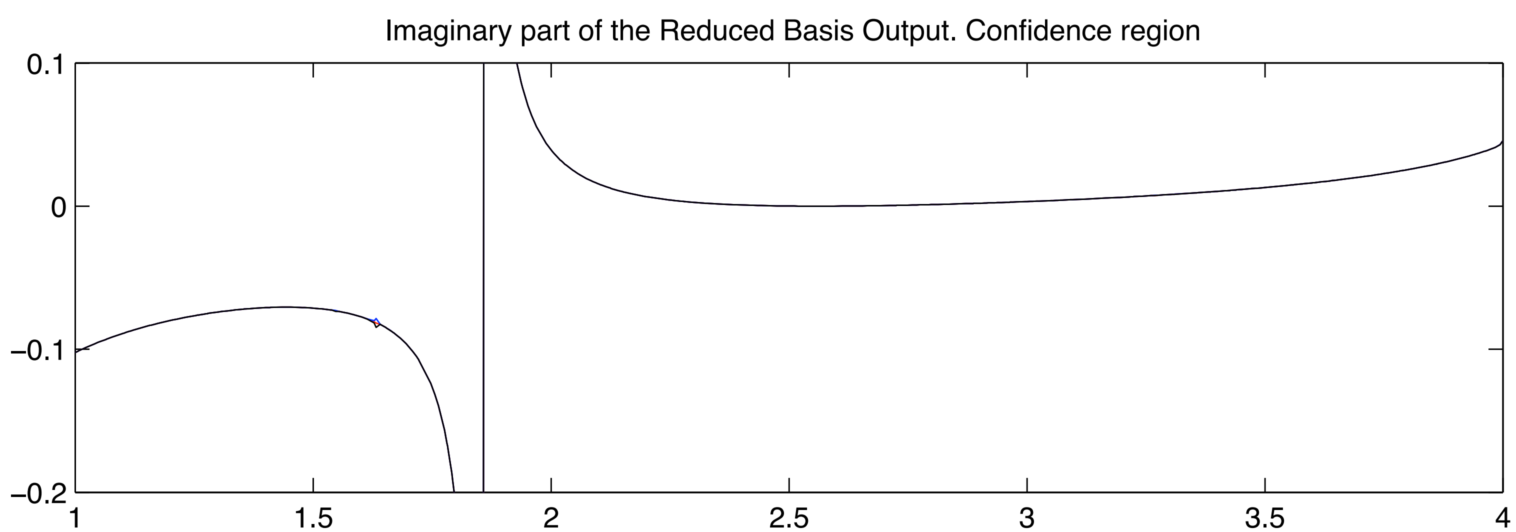


0.25 sec

N=12



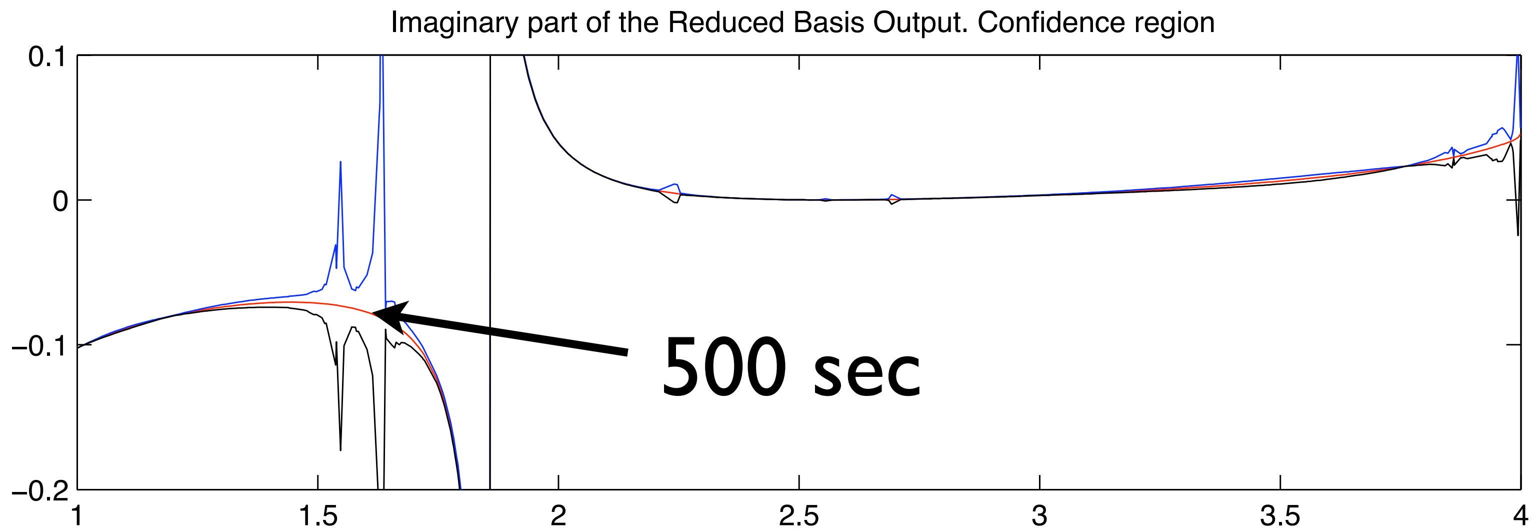
N=15



Lets see if it works

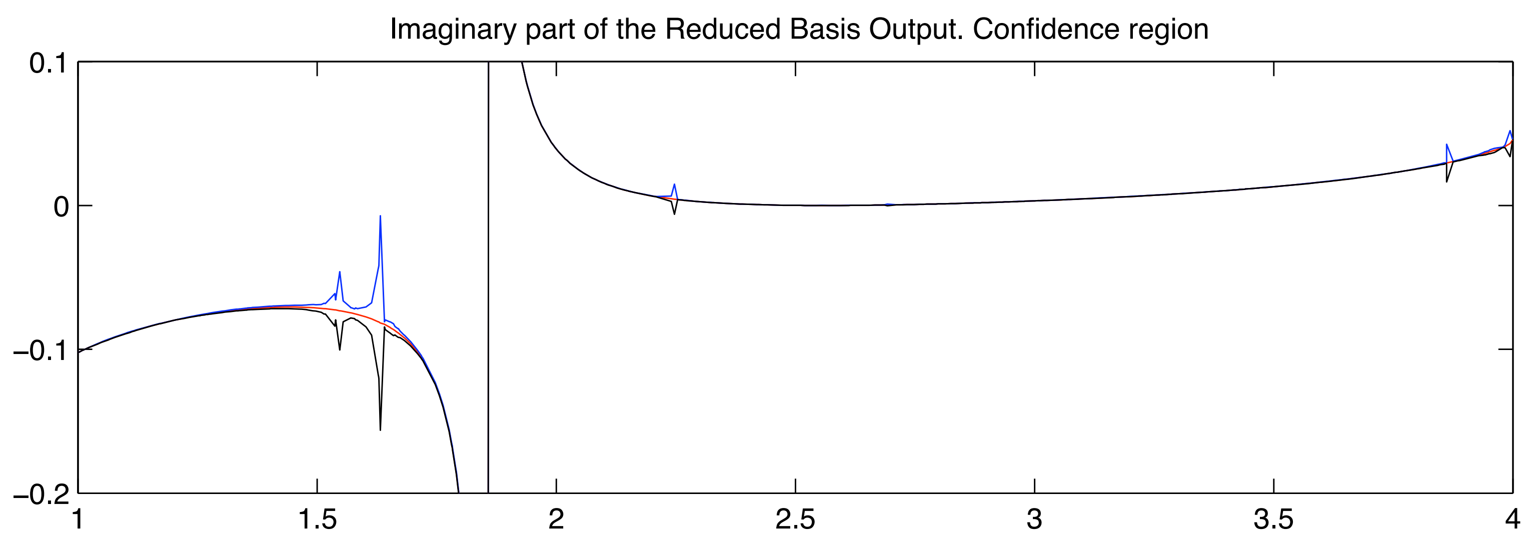
ÉCOLE POLYTECHNIQUE FÉDÉRALE DE LAUSANNE

N=10



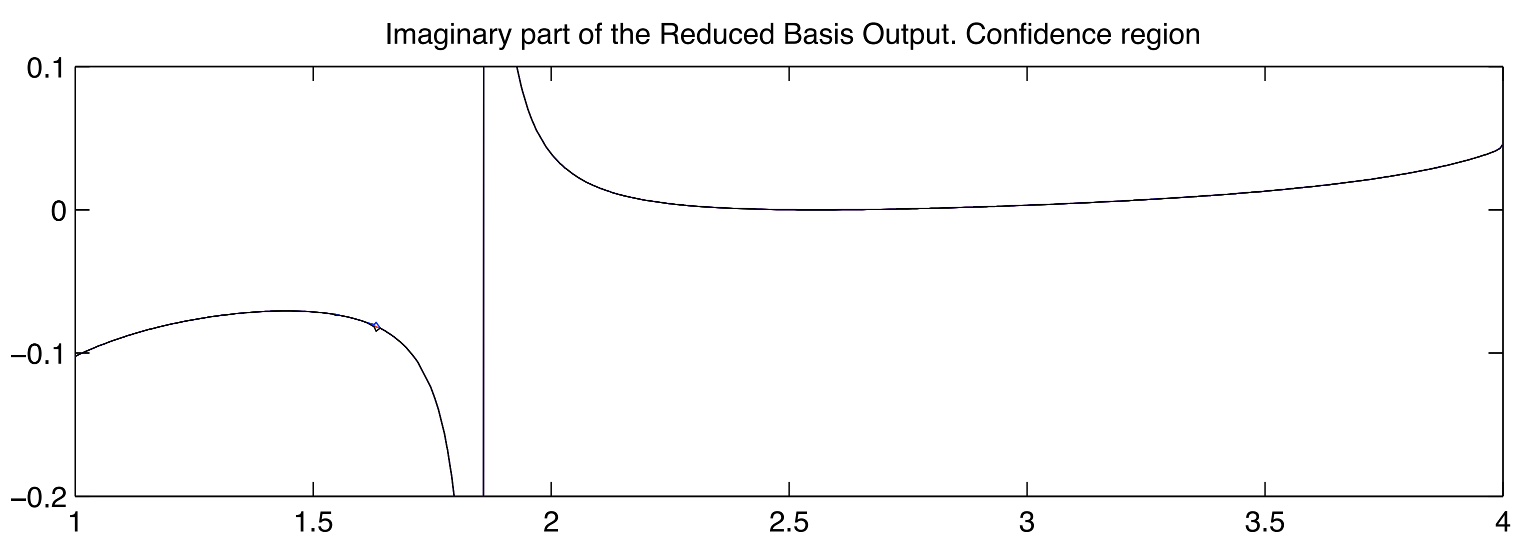
0.25 sec

N=12



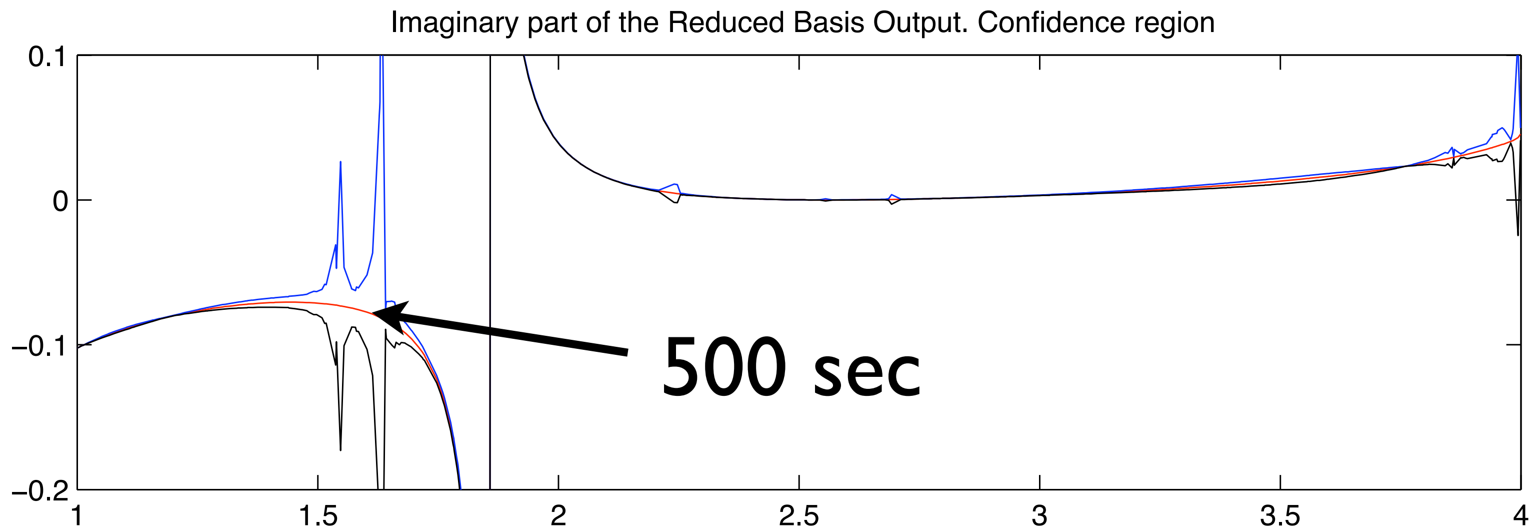
0.28 sec

N=15



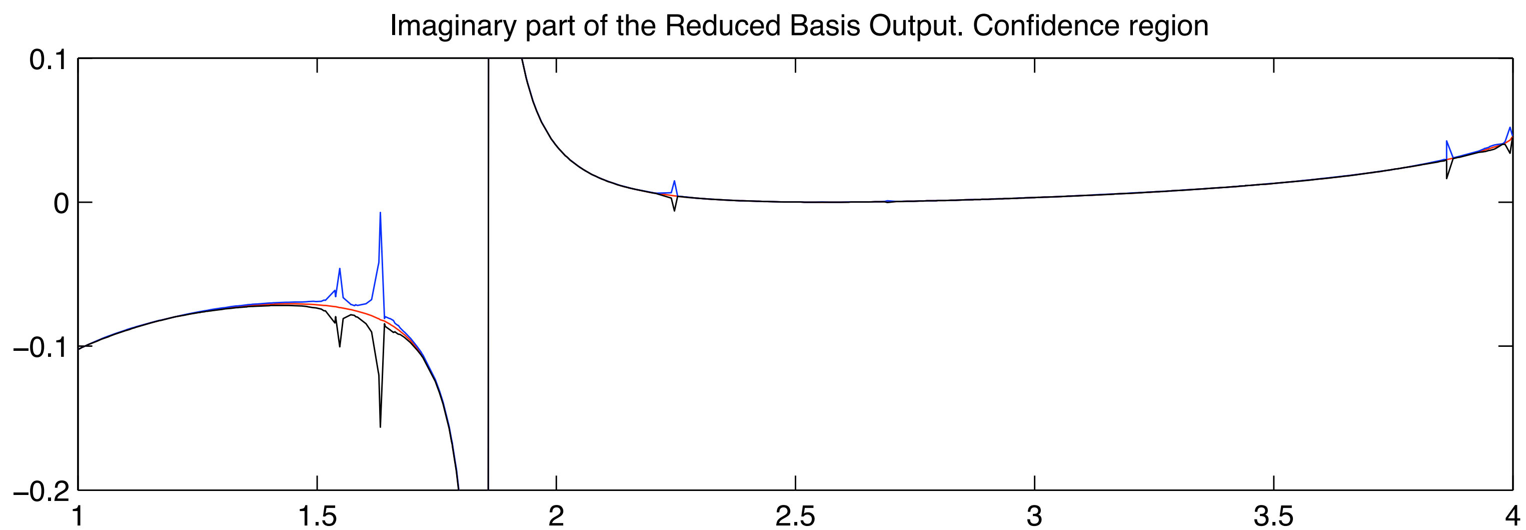
Lets see if it works

N=10



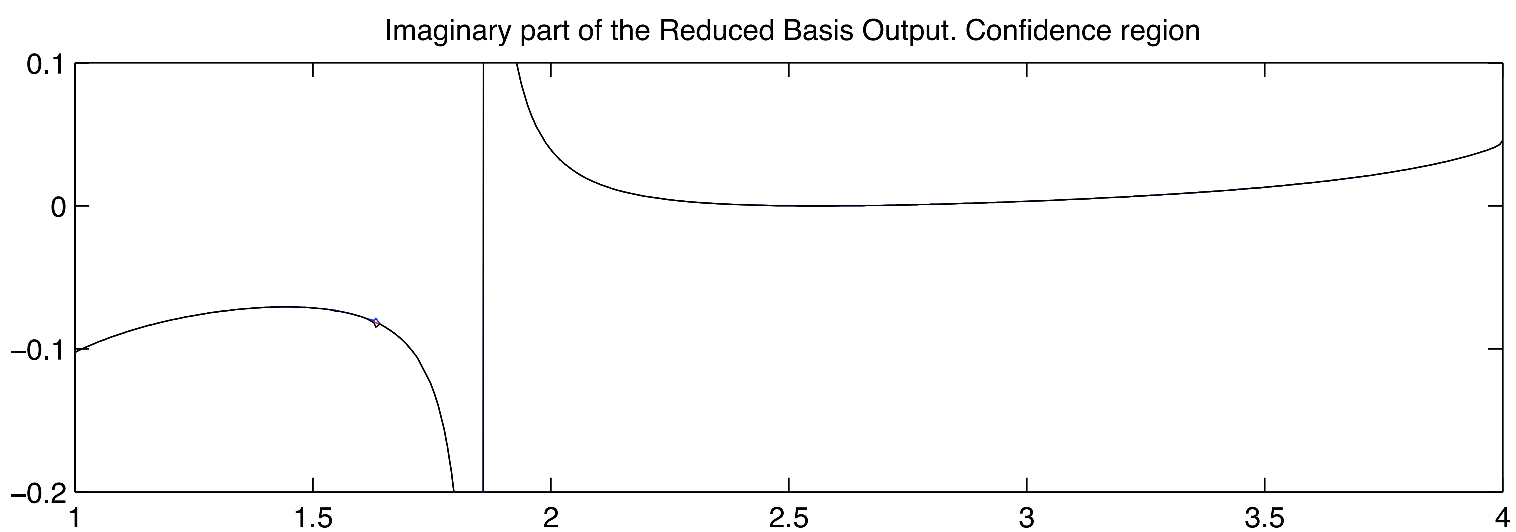
0.25 sec

N=12



0.28 sec

N=15



0.35 sec

An acoustic horn problem

Geometry Parameters:

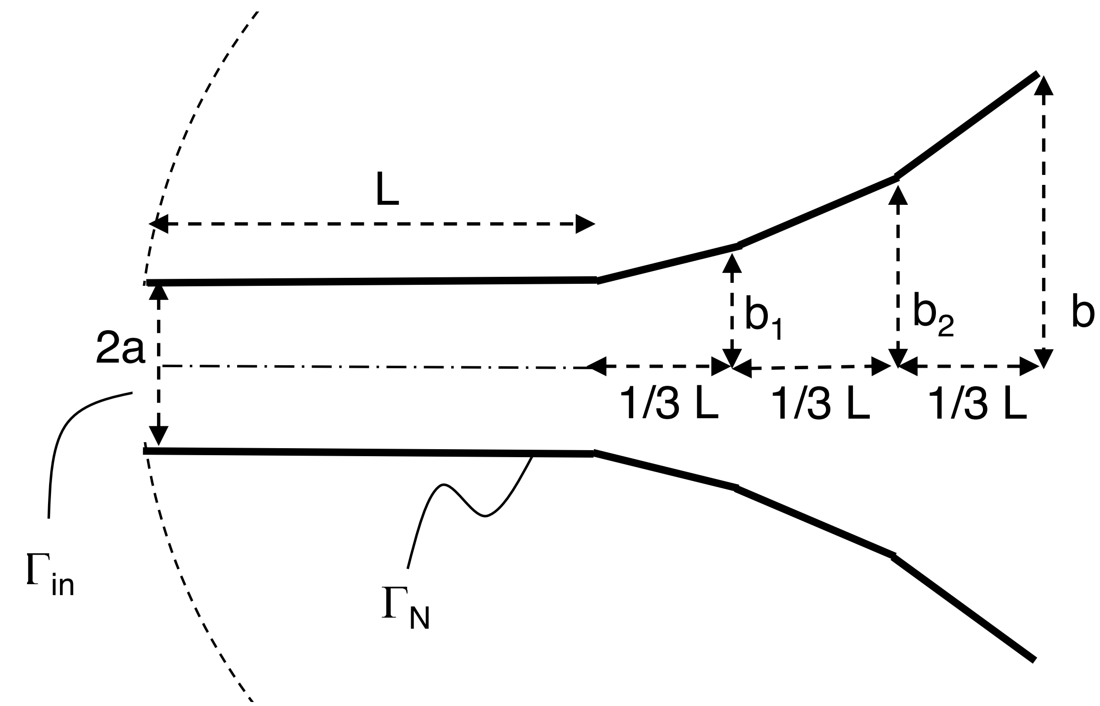
L = Length of waveguide and flare

$2a$ = waveguide width

b_1 = height of 1st slice of flare from center

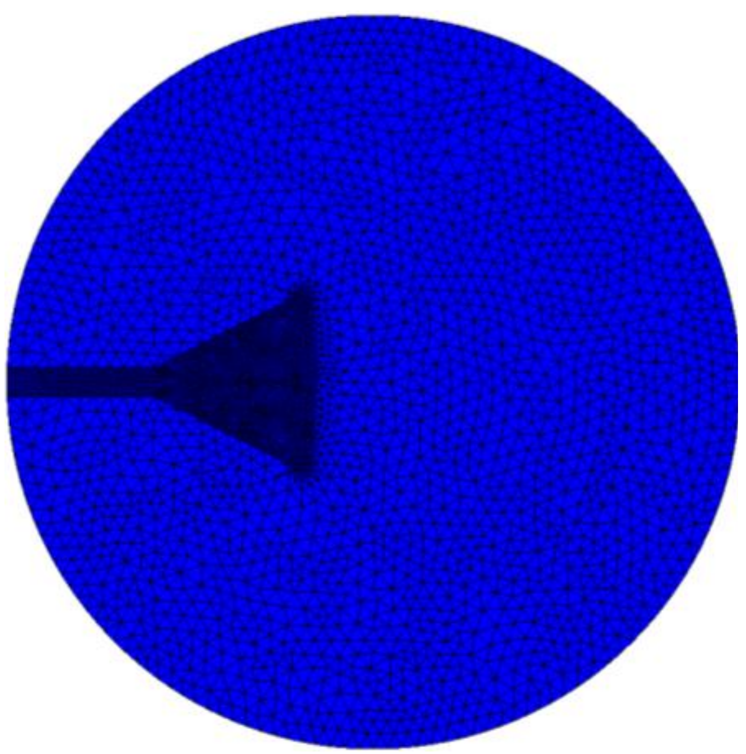
b_2 = height of 2nd slice of flare from center

b = final height of flare from center



Governing Equation: Helmholtz equation

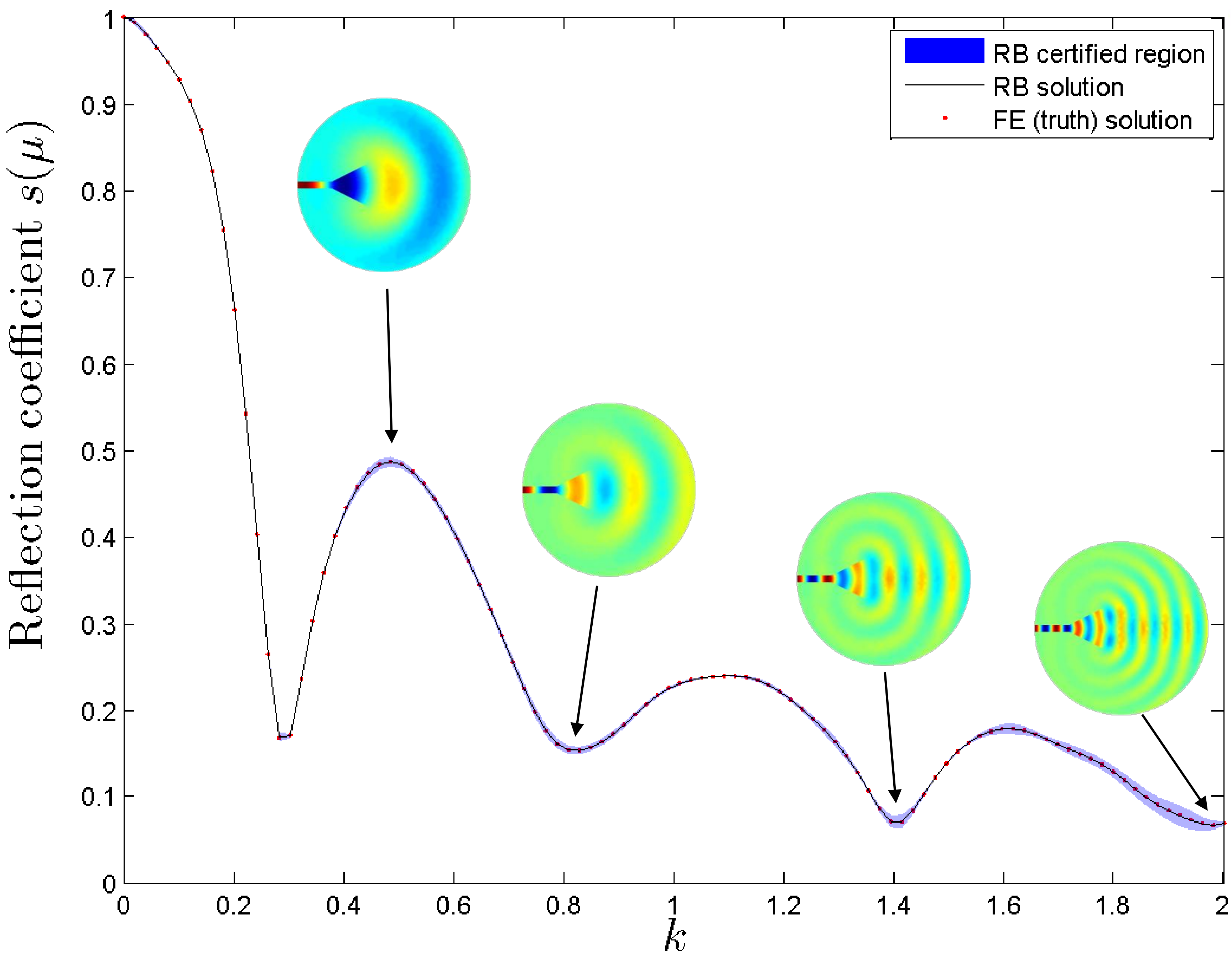
$$\nabla^2 u + k^2 u = 0,$$

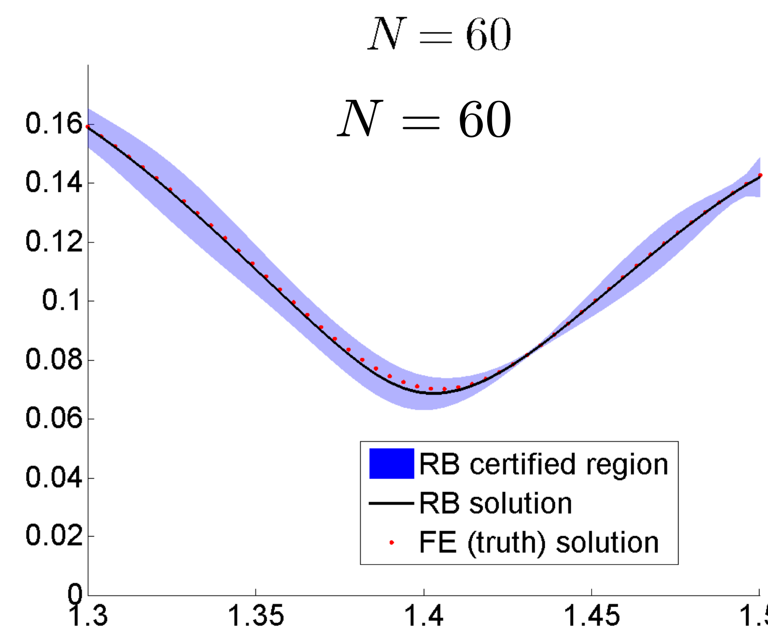


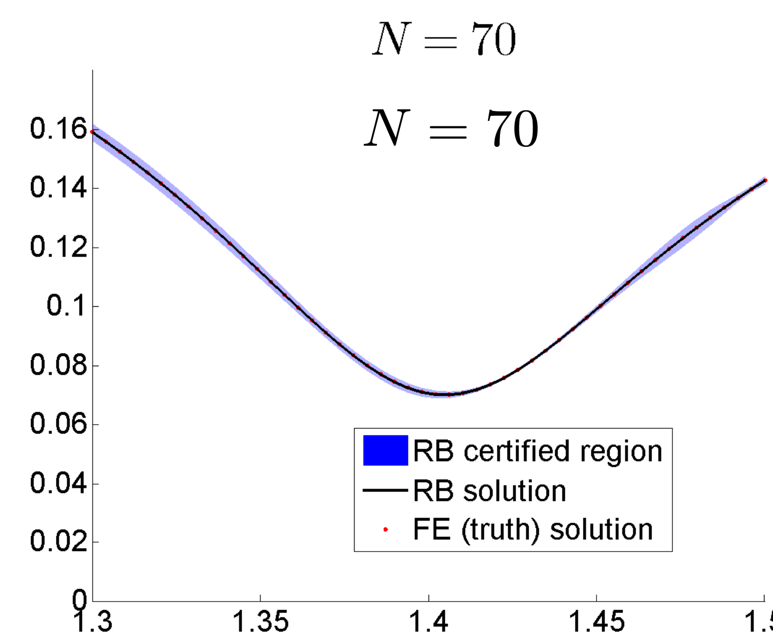
By A. Patera and K. Willcox

An acoustic horn problem

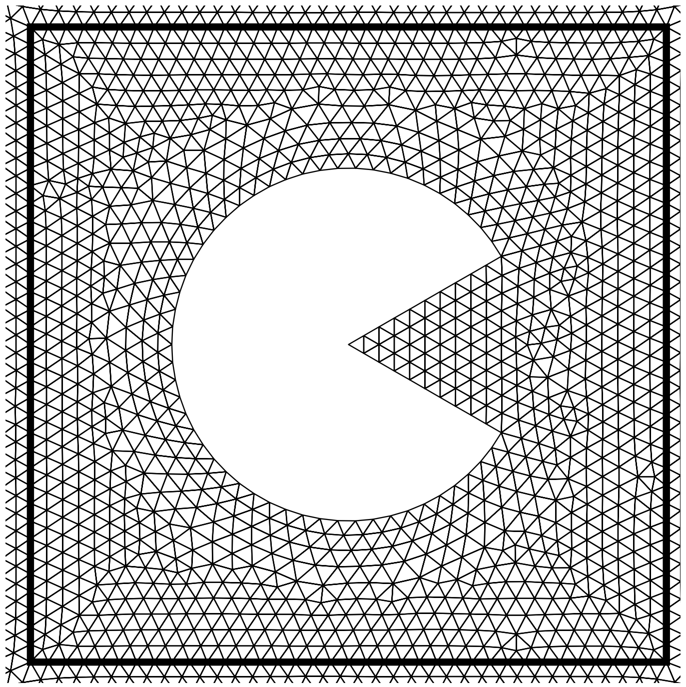
Wavenumber is parameter. N=150





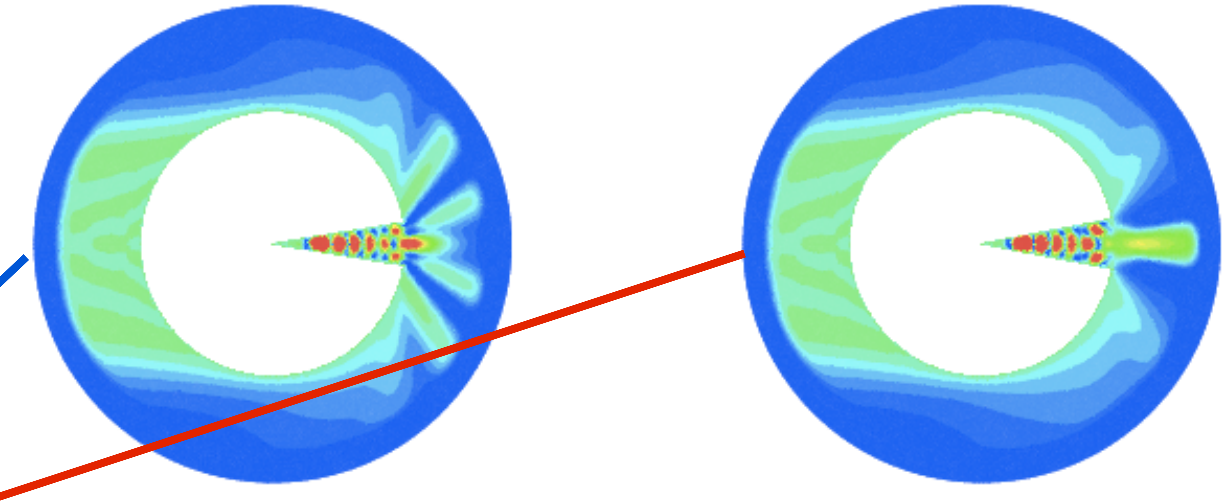


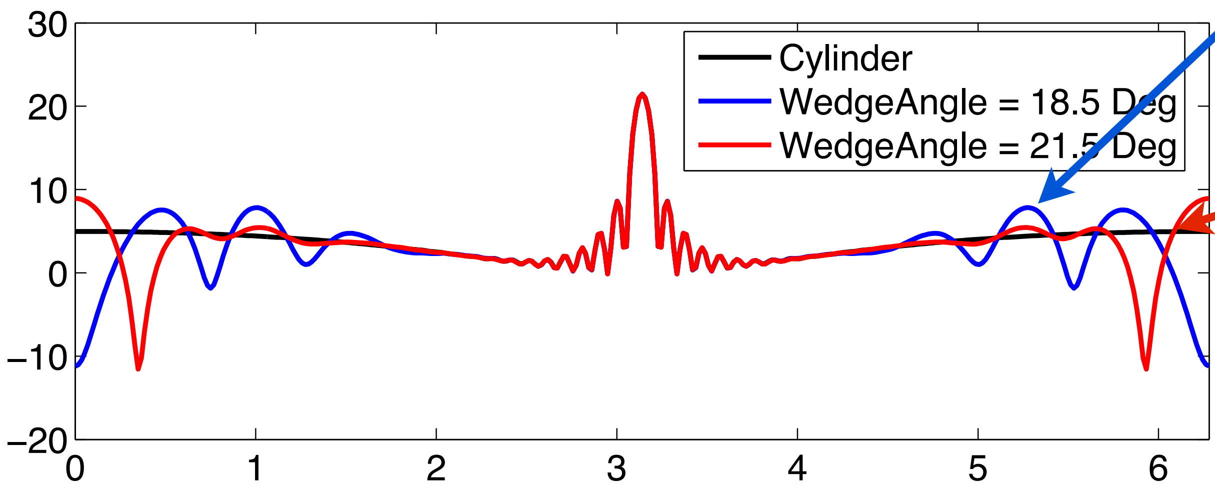
2D Pacman problem



Scattering by 2D PEC Pacman

Backscatter depends very sensitively on cutout angle and frequency.

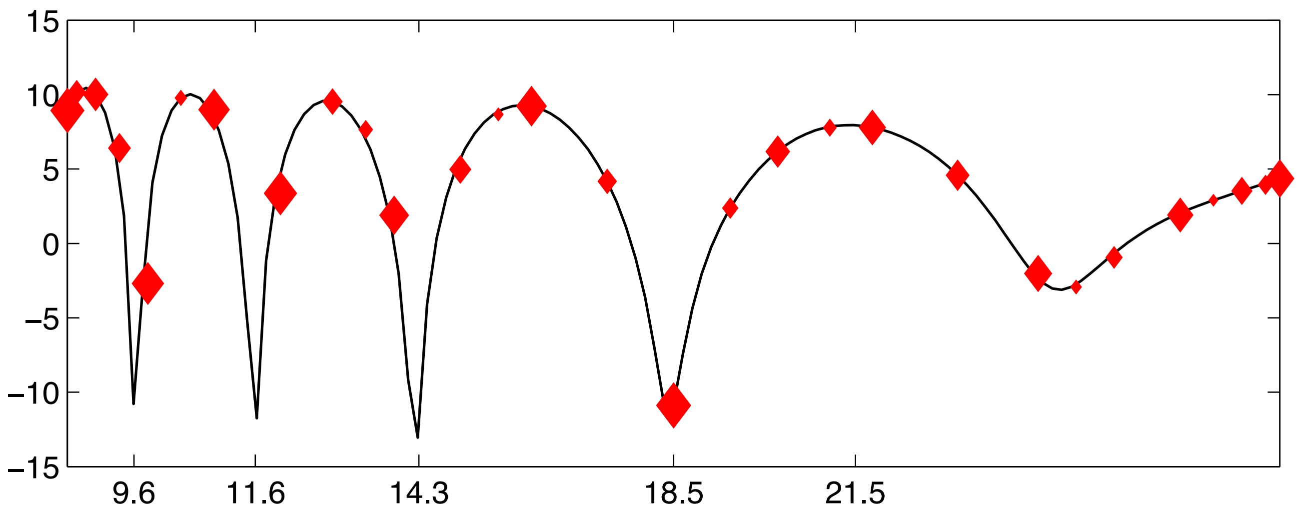




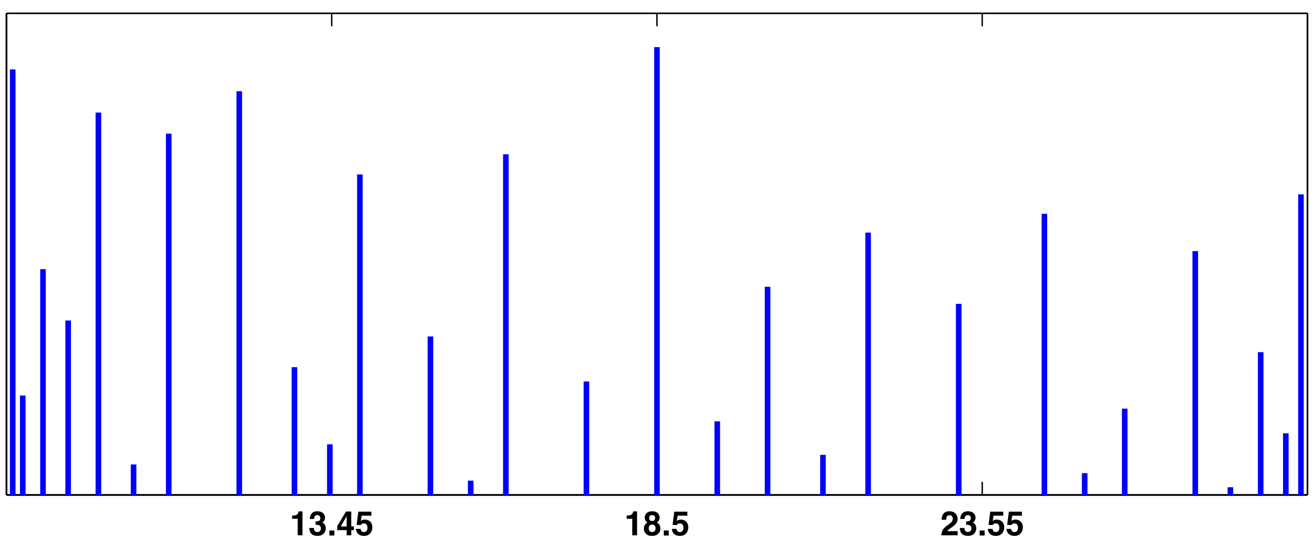
TM polarization

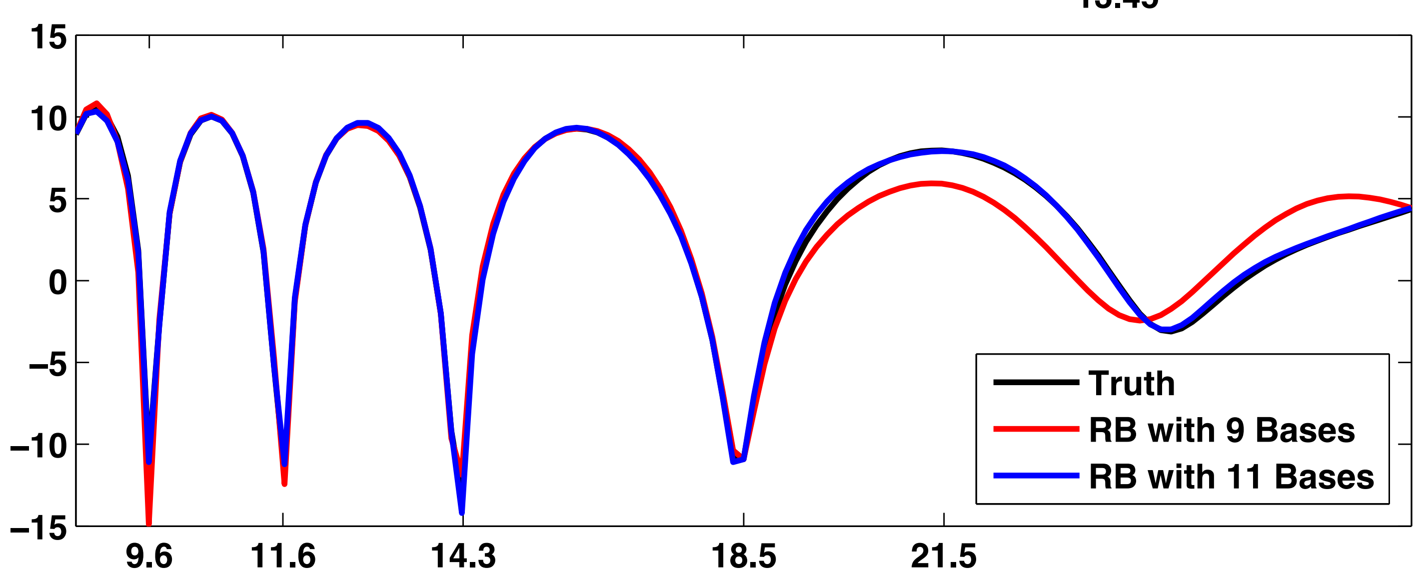
Difference in scattering is clear in fields

2D Pacman problem



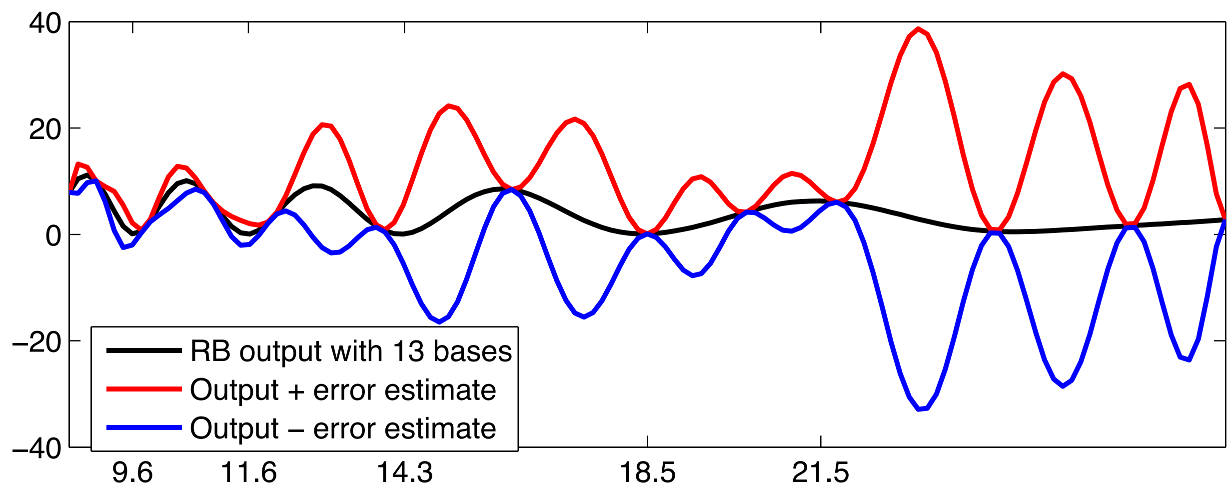
Greedy approach selects critical angles early in the selection process



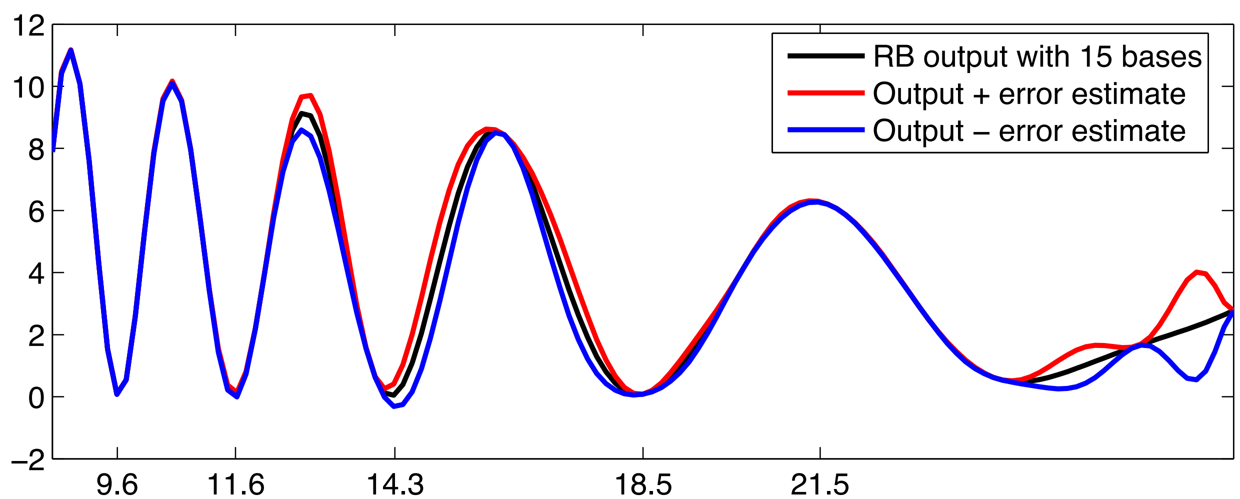


Convergence of output with O(10) basis elements

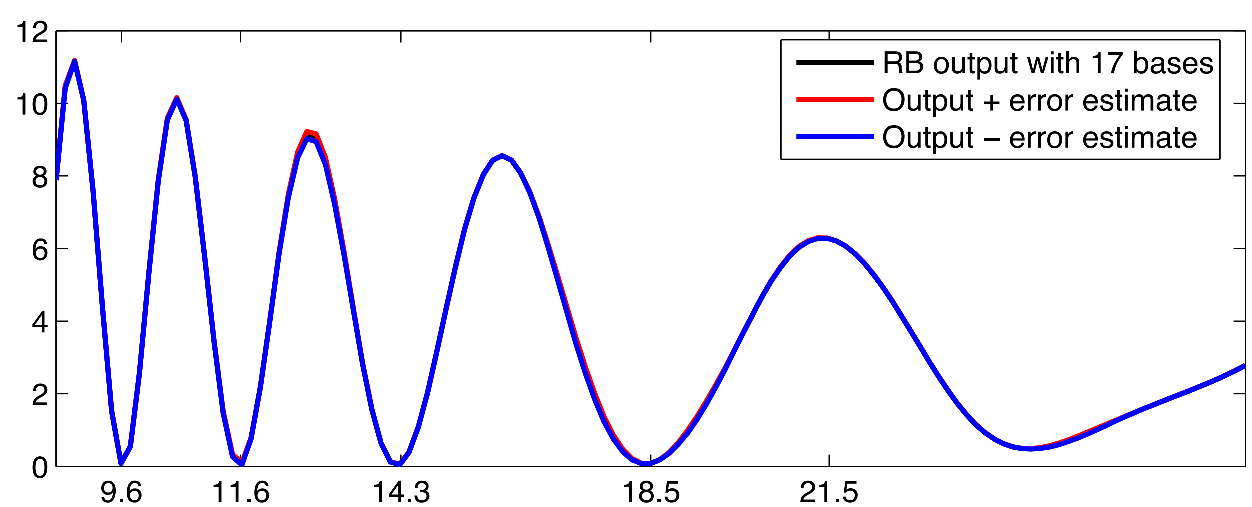
2D Pacman problem

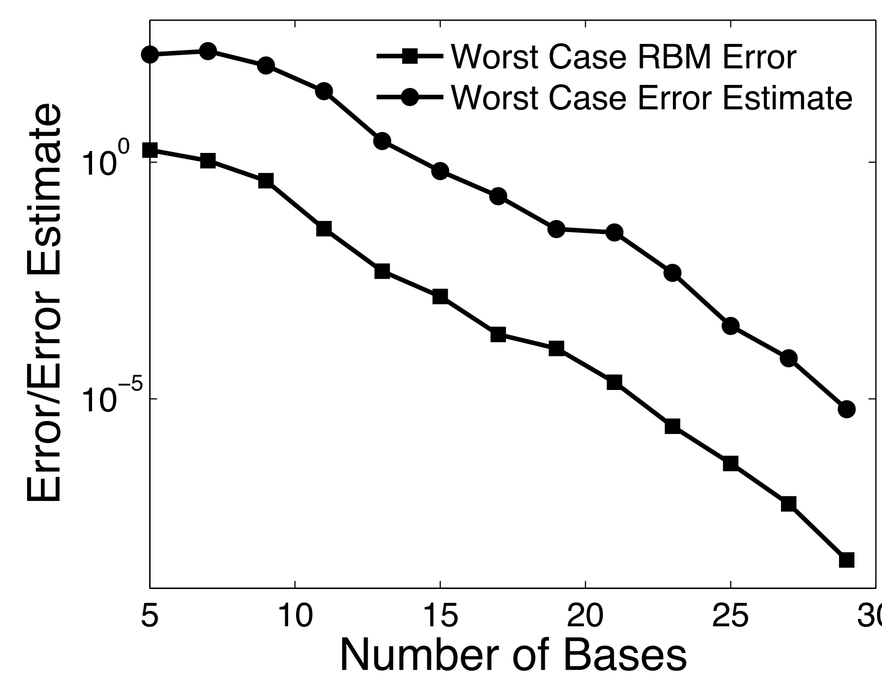


Convergence of error bounds over full parameter range



Exponential convergence of predicted error estimator and real error over large training set

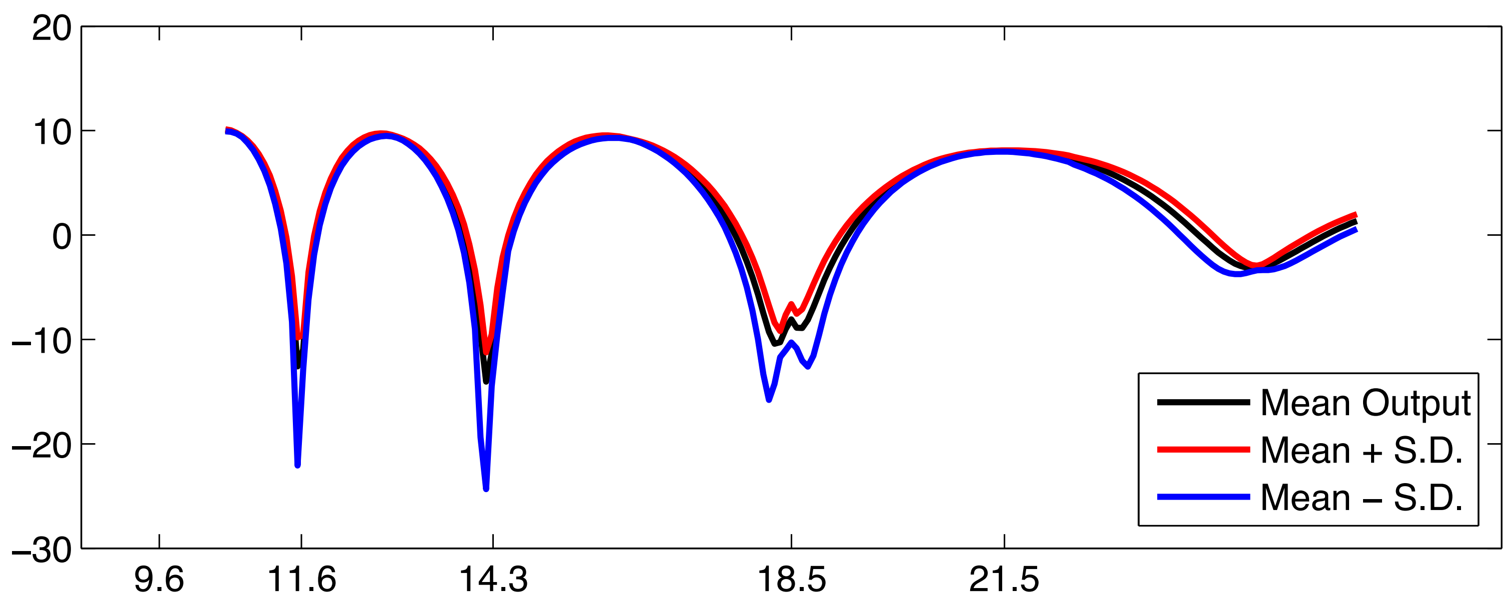


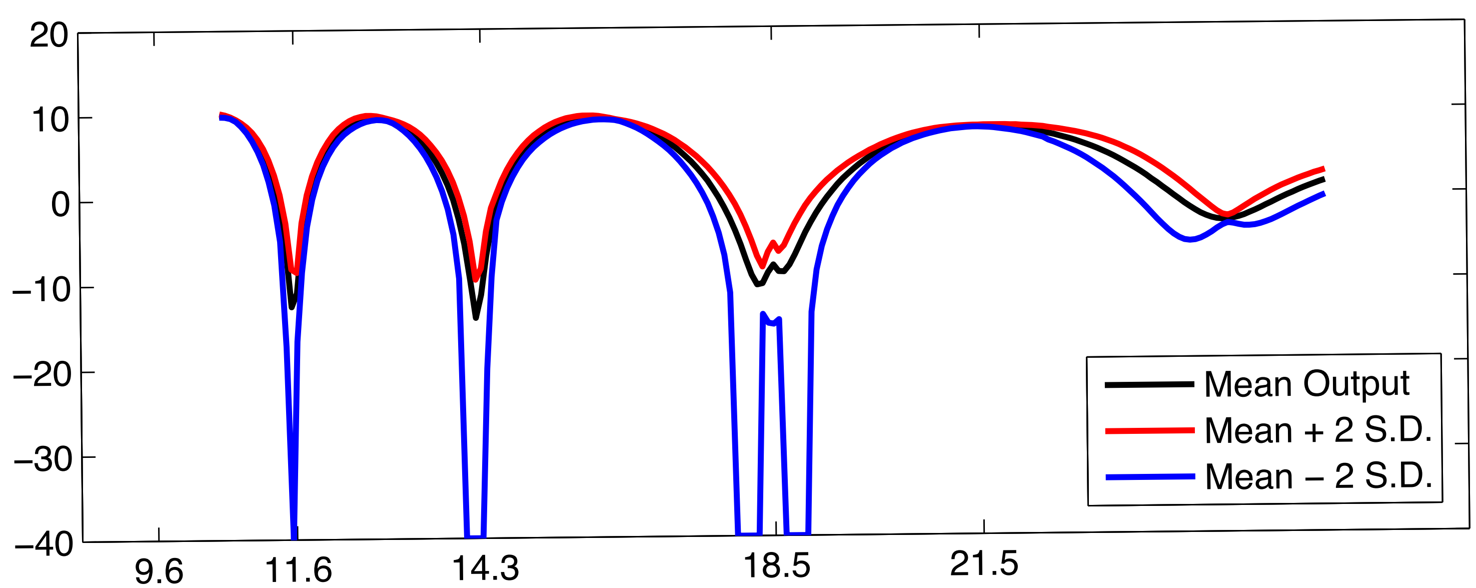


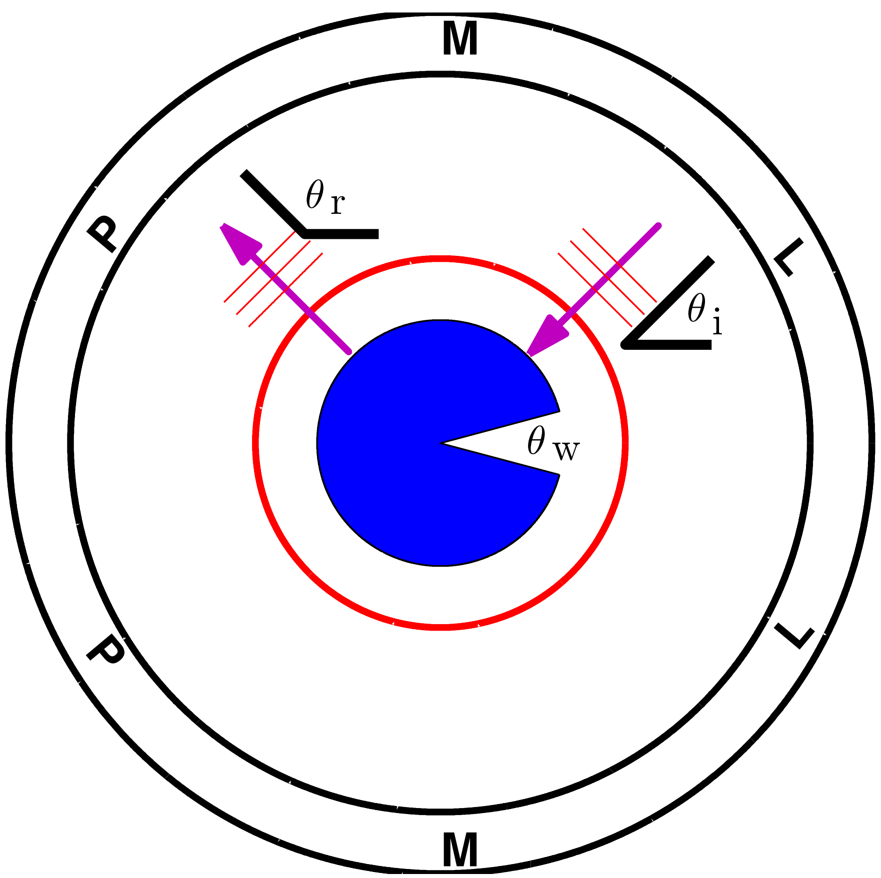
Note: Linear scale, not db scale

2D Pacman prototype for UQ

Fast evaluation over parameter space allows for rapid uncertainty quantification



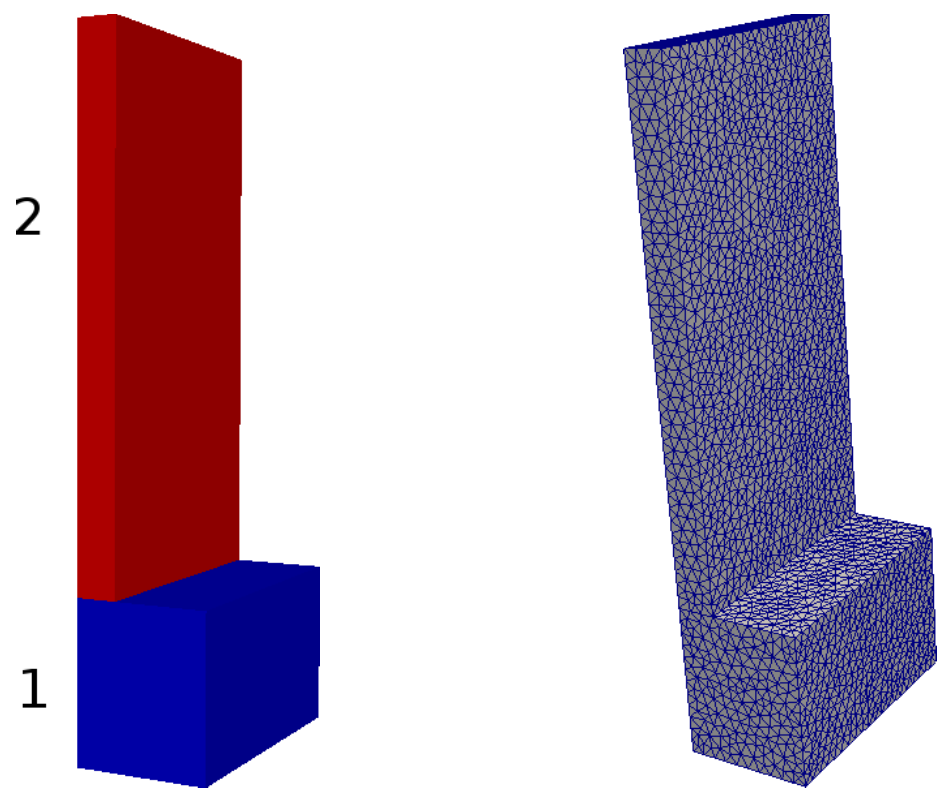




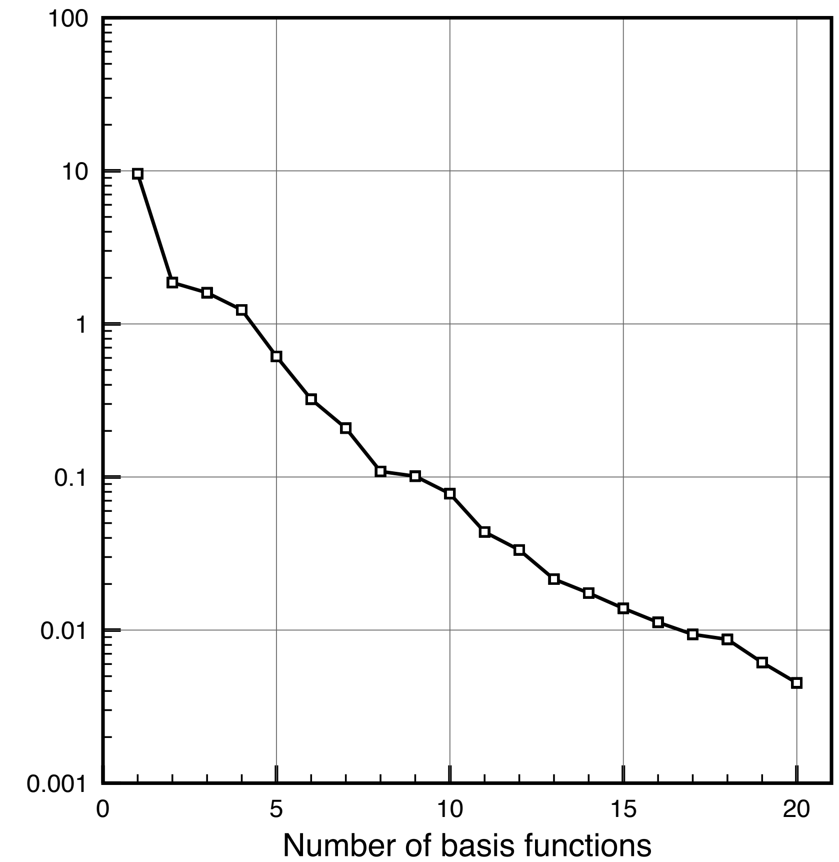
Uniformly distributed 5% randomness in gap angle

One last example

Thermal fin optimization



(a) (b)



Three parameters

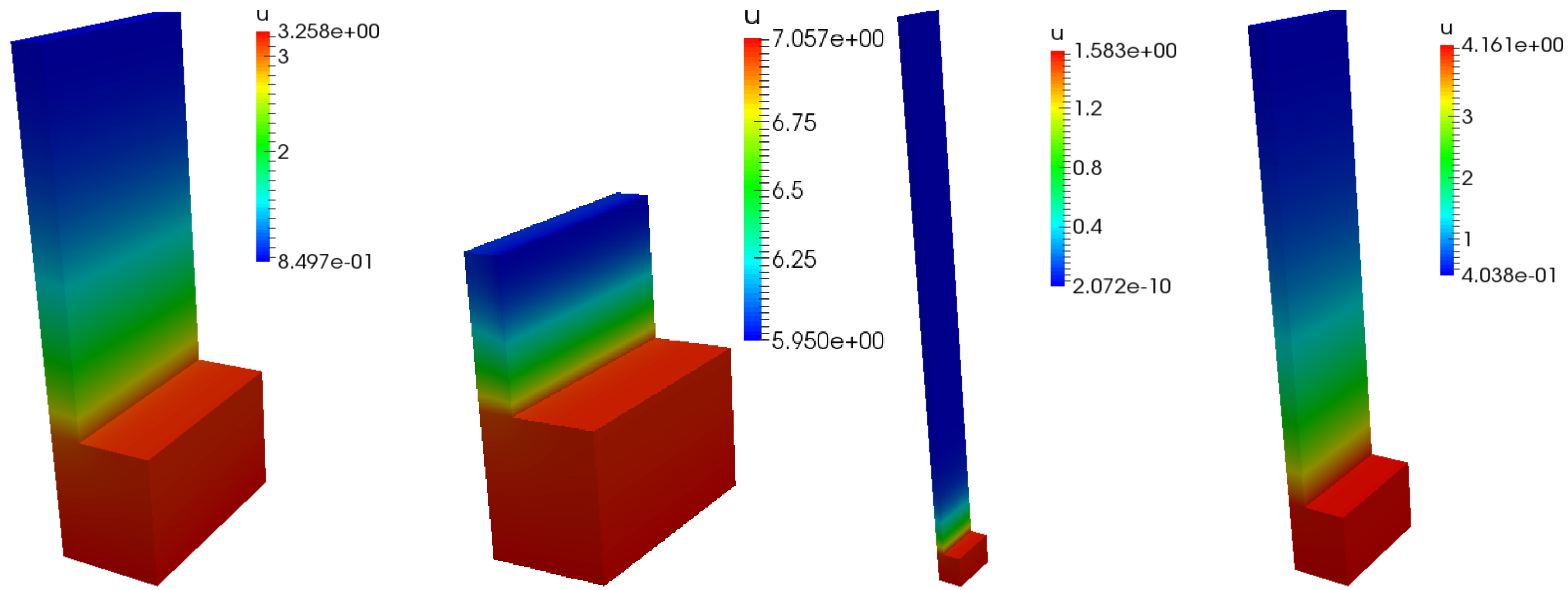
$$\mu_{[1]} \in [0.1, 1.0], \qquad \mu_{[2]} \in [0.5, 10.0], \qquad \mu_{[3]} \in [1.0, 10.0].$$

$$a_\circ(w, v; \mu) = \mu_{[3]} \int_{\Omega_\circ^1} \nabla w \cdot \nabla v + \int_{\Omega_\circ^2(\mu)} \nabla w \cdot \nabla v + \mu_{[1]} \int_{\Gamma_1} w\, v,$$

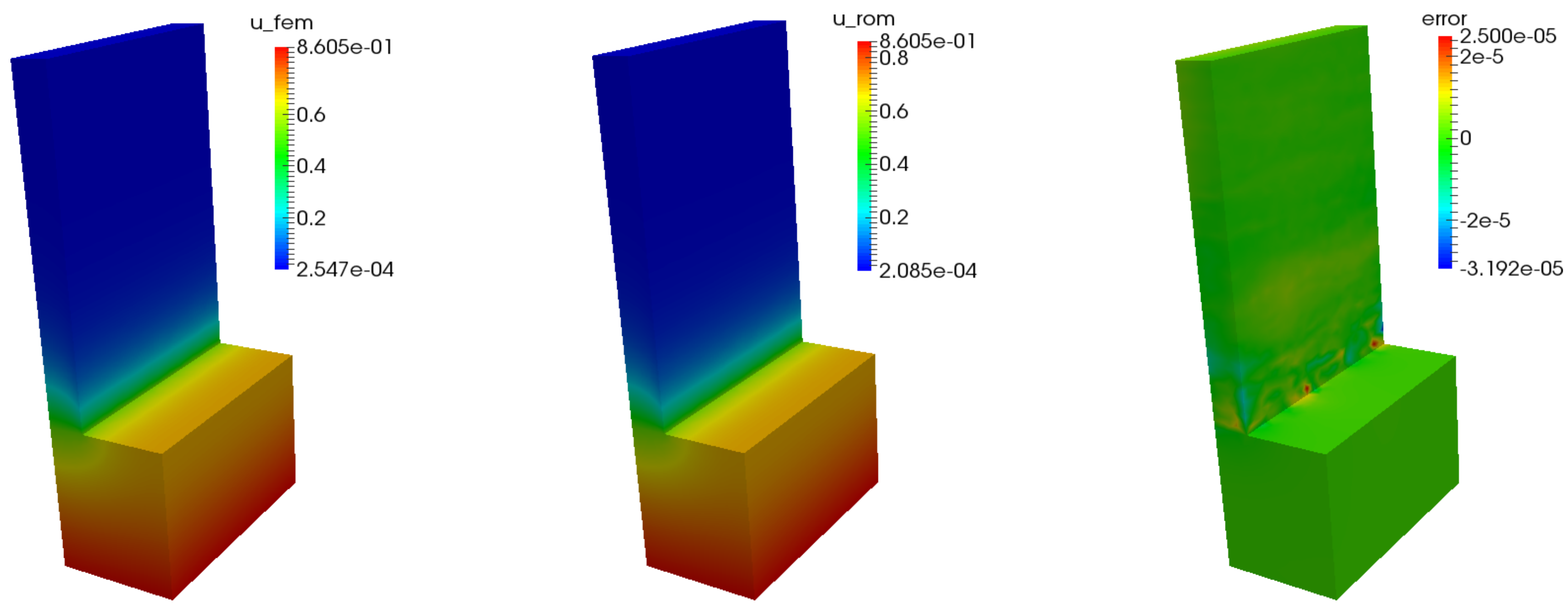
$$f_\circ(v) = \int_{\Gamma_{\text{bottom}}} v,$$

One last example

Basis functions



Verification



(a) (b) (c)

Where are we now ?

So far we have established

- Method works and efficient though greedy
- Performance depends on N
- Error is certified

Still to consider the 'non's'

- Non-affine problems
- Non-compliant problems
- Non-intrusive problems
- Non-stationary problems
- Non-standard problems

Lecture 3

Questions ?