Reduced basis methods for the resolution of parameter-dependent PDEs
MS13

Elise Grosjean
Ensta-Paris

Ensta-Paris
Institut Polytechnique de Paris

1. Reminders
   - Reduced Basis Methods
   - aposteriori errors

Reduced Basis Methods (RBM)

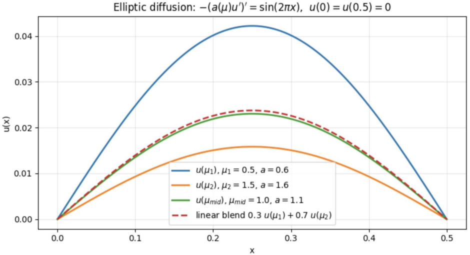
Solution manifold

HF solution manifold:

$$\mathcal{M}_h = \{u_h(\boldsymbol{\mu}) |\ \boldsymbol{\mu} \in\ \mathcal{G}\}$$

How can we reduce the manifold complexity?

- ◇ Keep HF precision
- ◇ Reduce computational costs



Reduced space

V^N

Kolmogorov N-width $=$ error from the **linear** space that best fit the solution manifold

$$V_N = \mathrm{Span}\{u_1, u_2, \dots u_N\}$$

where $u_1, \dots, u_N =$ snapshots

TP RB+FEM

- **Offline** Construction of a reduced space V_N spanned by a reduced basis $\rightarrow$???
- **Online** Computation of the reduced coefficients α $\rightarrow$???

TP RB+FEM

- **Offline** Construction of a reduced space V_N spanned by a reduced basis $\rightarrow$???
  $V_M = Span\{u_1, \dots, u_M\}$ (or $V_{Ntrain} = Span\{u_1, \dots, u_{Ntrain}\}$)
  Instead of approximate $u(\mathbf{x}, \boldsymbol{\mu})$ by $\sum_{k=1}^{M} \alpha_k(\boldsymbol{\mu}) u_k(\mathbf{x})$, we take $\sum_{k=1}^{N} a_k(\boldsymbol{\mu}) \Phi_k(\mathbf{x})$ with $N < M$.

$$\boxed{\min_{\|\Phi_1\|=1} \mathbb{E}[\|u - (u, \Phi_1)\Phi_1\|^2] \text{ or } C\Phi_1 = \lambda_1 \Phi_1}$$

- **Online** Computation of the reduced coefficients $\alpha \rightarrow$???

TP RB+FEM

- **Offline** Construction of a reduced space V_N spanned by a reduced basis $\rightarrow$???
  $V_M = Span\{u_1, \dots, u_M\}$ (or $V_{Ntrain} = Span\{u_1, \dots, u_{Ntrain}\}$)
  Instead of approximate $u(\mathbf{x}, \boldsymbol{\mu})$ by $\sum_{k=1}^{M} \alpha_k(\boldsymbol{\mu}) u_k(\mathbf{x})$, we take $\sum_{k=1}^{N} a_k(\boldsymbol{\mu}) \Phi_k(\mathbf{x})$ with $N < M$.

$$\boxed{\min_{\|\Phi_1\|=1} \mathbb{E}[\|u - (u, \Phi_1)\Phi_1\|^2] \text{ or } C\Phi_1 = \lambda_1 \Phi_1}$$

```
Construct_RB(NumberOfSnapshots=100,Nx=50,Ny=50,NumberOfModes=20)
```

- **Online** Computation of the reduced coefficients $\alpha \rightarrow$???

TP RB+FEM

- **Offline** Construction of a reduced space V_N spanned by a reduced basis $\rightarrow$???
  $V_M = Span\{u_1, \dots, u_M\}$ (or $V_{Ntrain} = Span\{u_1, \dots, u_{Ntrain}\}$)
  Instead of approximate $u(\mathbf{x}, \boldsymbol{\mu})$ by $\sum_{k=1}^{M} \alpha_k(\boldsymbol{\mu}) u_k(\mathbf{x})$, we take $\sum_{k=1}^{N} a_k(\boldsymbol{\mu}) \Phi_k(\mathbf{x})$ with $N < M$.

  $$\boxed{\min_{\|\Phi_1\|=1} \mathbb{E}[\|u - (u, \Phi_1)\Phi_1\|^2] \text{ or } C\Phi_1 = \lambda_1 \Phi_1}$$

  ```
  Construct_RB(NumberOfSnapshots=100,Nx=50,Ny=50,NumberOfModes=20)
  ```
- **Online** Computation of the reduced coefficients $\boldsymbol{\alpha} \rightarrow$???

TP RB+FEM

- **Offline** Construction of a reduced space V_N spanned by a reduced basis $\rightarrow$???
  $V_M = Span\{u_1, \dots, u_M\}$ (or $V_{Ntrain} = Span\{u_1, \dots, u_{Ntrain}\}$)
  Instead of approximate $u(\mathbf{x}, \boldsymbol{\mu})$ by $\sum_{k=1}^{M} \alpha_k(\boldsymbol{\mu}) u_k(\mathbf{x})$, we take $\sum_{k=1}^{N} a_k(\boldsymbol{\mu}) \Phi_k(\mathbf{x})$ with $N < M$.

$$\boxed{\min_{\|\Phi_1\|=1} \mathbb{E}[\|u - (u, \Phi_1)\Phi_1\|^2] \text{ or } C\Phi_1 = \lambda_1 \Phi_1}$$

```
Construct_RB(NumberOfSnapshots=100,Nx=50,Ny=50,NumberOfModes=20)
```

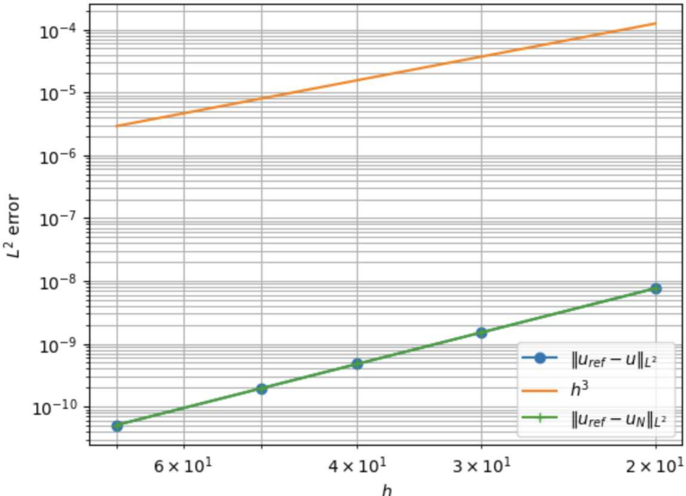
- **Online** Computation of the reduced coefficients $\boldsymbol{\alpha} \rightarrow$???
  $\mathbf{P}^T \mathbf{A}(\boldsymbol{\mu}) \mathbf{P}\, \boldsymbol{\alpha}(\boldsymbol{\mu}) = \mathbf{P}^T \mathbf{l}(\boldsymbol{\mu})$

```
solve_fem_rom(A,b,mu, Phi,m)
```

TP RB+FEM

Expected results: Convergence in $\mathcal{O}(h^3)$!!

$\mu = 0.6$



TP RB+FEM

- ◇ Keep HF precision $\surd$
- ◇ Reduce computational costs ?

TP RB+FEM

```
import time
start = time.time()
# code blabla
end = time.time()
print("Time :", end - start, "secondes")
```

TP RB+FEM

```
def FEMsolve(A0,b,m,mu):

 # solve
 A= A_mu(mu)*A0
 A, b = enforce(A, b, D=m.boundary_nodes())
 u = solve(A, b)
 return u
```

Solve $\mathbf{Au} = \mathbf{l}$ with $\mathbf{A}(\boldsymbol{\mu}) = \sum_q \theta_q^a(\boldsymbol{\mu})\mathbf{A}_q$ and $\mathbf{l}(\boldsymbol{\mu}) = \sum_q \theta_q^l(\boldsymbol{\mu})\mathbf{l}_q$???

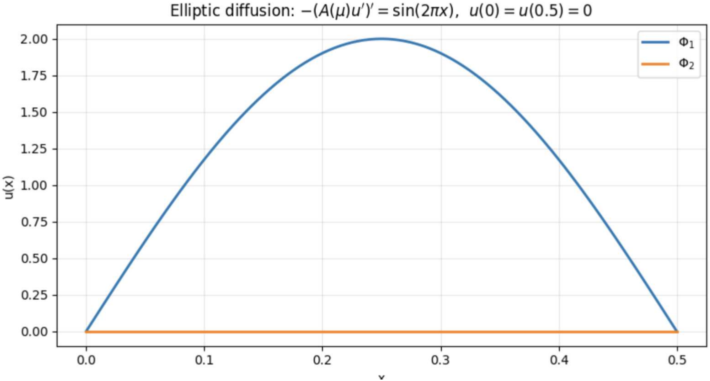
$$\mathbf{A}(\mu) = \underbrace{A\_mu(\mu)}_{0.1+\mu}\mathbf{A}_0$$

TP RB+FEM

- ◇ Keep HF precision $\surd$
- ◇ Reduce computational costs ?

TP RB+FEM

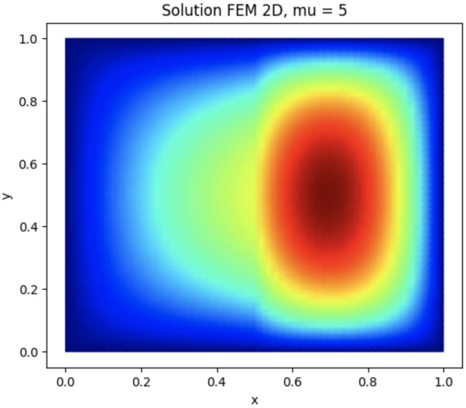
One can show that the more regularizing the operator C is, the faster its eigenvalues decay!



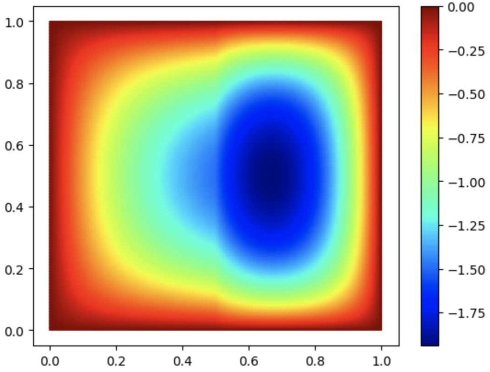
TP Kolmogorov

TP4: Complexify the problem with

- ◇ 1D → 2D ($[0,1] \times [0,1]$)
- ◇ Diffusion coefficient $A(\mathbf{x}, \mu) = \begin{cases} \mu \in \Omega_1 = \{(x,y) \in \Omega,\ \ x < 0.5\} \\ 1 \in \Omega_2 = \Omega \backslash \Omega_1 \end{cases}$

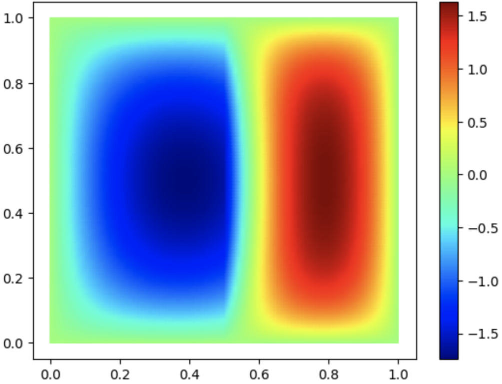


TP Kolmogorov



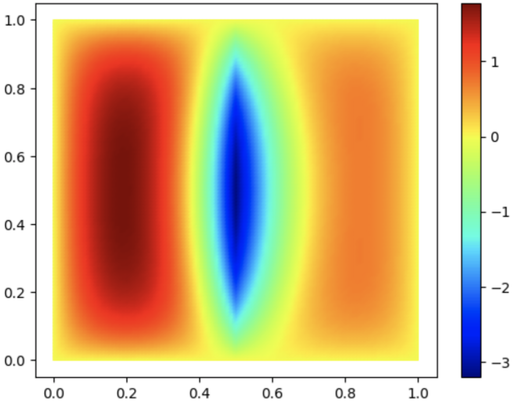
mode 1

TP Kolmogorov



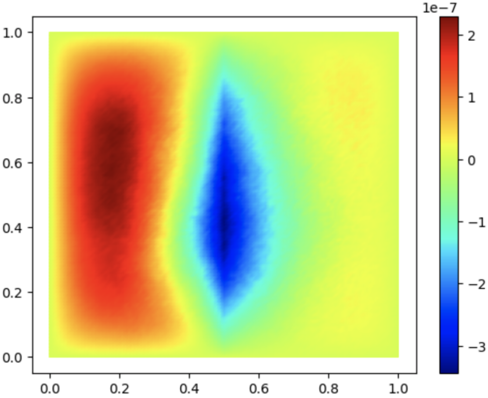
mode 2

TP Kolmogorov



mode 3

TP Kolmogorov



mode 4

Computable error bound

$$\|u_h(\mu_m) - P_N(u_h(\mu_m))\|_V^2 \leq \sum_{i=1}^{M} \|u_h(\mu_i) - P_N(u_h(\mu_i))\|_V^2 = M \sum_{k=N+1}^{M} \lambda_k$$

What about $\|u_h(\mu) - P_N(u_h(\mu))\|_V$ for $\mu \notin \{\mu_m\}_{1\leq m\leq M}$????

or

$\|u_h(\mu) - u_N(\mu)\|_V$ for any μ, including $\mu \in \{\mu_m\}_{1\leq m\leq M}$, since in practice the RB approximation is not optimal, i.e., $u_N(\mu) \neq P_N(u_h(\mu))$????

Computable error bound

$$\|u_h(\mu_m) - P_N(u_h(\mu_m))\|_V^2 \leq \sum_{i=1}^{M} \|u_h(\mu_i) - P_N(u_h(\mu_i))\|_V^2 = M \sum_{k=N+1}^{M} \lambda_k$$

We want a bound that depends only on the RB approximation u_N :

aposteriori bound !!!

Aposteriori

We are going to use $\|u_h(\boldsymbol{\mu}) - u_N(\boldsymbol{\mu})\|_V \leq \frac{1}{\alpha_{sta}(\boldsymbol{\mu})}\|r_N(\boldsymbol{\mu})\|_{V'}$

"A posteriori numerical analysis based on the method of equilibrated fluxes", M. Vohralik

https://who.rocq.inria.fr/Martin.Vohralik/Enseig/APost/a$_p$*osteriori.pdf*

Aposteriori

An *a posteriori error estimator* is a function $\Delta_N : \mathcal{G} \to \mathbb{R}^+$ satisfying the following properties:

Robustness:

$$\forall \boldsymbol{\mu} \in \mathcal{G}, \qquad \|u_h(\boldsymbol{\mu}) - u_N(\boldsymbol{\mu})\|_V \leq \Delta_N(\boldsymbol{\mu}).$$

Efficiency :

$$\forall \boldsymbol{\mu} \in \mathcal{G}, \quad \exists K(\boldsymbol{\mu}) > 0 \quad \text{such that} \quad \Delta_N(\boldsymbol{\mu}) \leq K(\boldsymbol{\mu})\|u_h(\boldsymbol{\mu}) - u_N(\boldsymbol{\mu})\|_V$$

Asymptotic exactness: the effectivity index $I_{eff} = \frac{\Delta_N(\boldsymbol{\mu})}{\|u_h - u_N\|} \underset{N\to\infty}{\to} 1$

Guaranteed upper bound: The function Δ_N can be evaluated for all $\boldsymbol{\mu} \in \mathcal{G}$ without evaluating $u_h(\boldsymbol{\mu})$ (fully computable from $u_N(\boldsymbol{\mu})$).

Small evaluation cost: Can be evaluated locally (only performing calculations in the element K or in its neighborhood $\mathcal{I}_K$)

Error components identification: Distinguish and estimate separately the different error components

Aposteriori

Two key ingredients:

- ◇ Dual norm of the residual: Offline-online computation strategy
- ◇ Inf-sup $\alpha_{sta}(\boldsymbol{\mu})$ not efficiently computable but one can compute $\alpha_{LB}(\boldsymbol{\mu})$ such that

$$\forall \boldsymbol{\mu} \in \mathcal{G}, \alpha_{sta}(\boldsymbol{\mu}) \geq \alpha_{LB}(\boldsymbol{\mu})$$

Case of coercivity: $\alpha^* = \inf_{v \neq 0} \frac{a(v,v)}{\|v\|_V^2}$

Remark: When $\alpha_{sta}(\boldsymbol{\mu})$ becomes small, $K(\boldsymbol{\mu}) = \frac{\gamma}{\alpha}$ becomes too big! Overestimate RB approximation.

Dual norm of the residual

Let's get back to our sheep

linear second-order parameter dependent problem

$$\begin{cases} -\nabla \cdot \big(a(\boldsymbol{\mu})\nabla u\big) = f(\boldsymbol{\mu}) & \text{dans } \Omega, \\ u = 0 & \text{sur } \partial\Omega. \end{cases}$$

$\alpha_{LB} = \alpha_{min}$

Dual norm of the residual

- Since in our setting $V = H_0^1(\Omega)$, thus $\|r\|_{V_h'} = r^T\mathbf{K}^{-1}r = (z,z)_{V_h} = z^T\mathbf{K}z$ where $\mathbf{K}$ is the stiffness matrix.
- Affine decomposition:

$$\mathbf{A}(\boldsymbol{\mu}) = \sum_q \theta_q^a(\boldsymbol{\mu})\mathbf{A}_q \quad \text{and} \quad \mathbf{l}(\boldsymbol{\mu}) = \sum_q \theta_q^l(\boldsymbol{\mu})\mathbf{l}_q$$

- Reduced system to solved:

$$\boxed{\mathbf{P}^T\mathbf{A}(\boldsymbol{\mu})\mathbf{P}\ \boldsymbol{\alpha}(\boldsymbol{\mu}) = \mathbf{P}^T\mathbf{l}(\boldsymbol{\mu}).}$$

Dual norm of the residual

$$
\begin{aligned}
\|\mathbf{A}(\boldsymbol{\mu})\mathbf{u_N}(\boldsymbol{\mu}) - \mathbf{l}(\boldsymbol{\mu})\|_{V'}^2 = &\sum_{q=1}^{Q^\ell}\sum_{k=1}^{Q^\ell} \theta_q^\ell(\boldsymbol{\mu})\theta_k^\ell(\boldsymbol{\mu}) \boxed{\mathbf{l_k}^T \mathbf{K}^{-1}\mathbf{l_q}} \\
&+ \sum_{q=1}^{Q^a}\sum_{k=1}^{Q^a} \theta_q^a(\boldsymbol{\mu})\theta_k^a(\boldsymbol{\mu})\, \boldsymbol{\alpha}(\boldsymbol{\mu})^T \boxed{\mathbf{P}^T\mathbf{A}_k^T\mathbf{K}^{-1}\mathbf{A}_q\mathbf{P}}\, \boldsymbol{\alpha}(\boldsymbol{\mu}) \\
&- 2\sum_{q=1}^{Q^\ell}\sum_{k=1}^{Q^a} \theta_q^\ell(\boldsymbol{\mu})\theta_k^a(\boldsymbol{\mu})\, \boldsymbol{\alpha}(\boldsymbol{\mu})^T \boxed{\mathbf{P}^T\mathbf{A}_k^T\mathbf{K}^{-1}\mathbf{l}_q}.
\end{aligned}
$$

Online complexity: since the boxed quantities are precomputed offline and $\alpha(\boldsymbol{\mu}) \in \mathbb{R}^N$ is known, the computational cost is

$$\mathcal{O}\left((Q^\ell)^2 + (Q^a)^2 N^2 + Q^\ell Q^a N\right).$$

We solve systems like $\mathbf{K}w = \mathbf{l_q}$ and $\mathbf{K}w = \mathbf{A_q P}$

Aposteriori

Two key ingredients:

- ◇ Dual norm of the residual: Offline-online computation strategy ☒
- ◇ Inf-sup $\alpha_{sta}(\boldsymbol{\mu})$ not efficiently computable but one can compute $\alpha_{LB}(\boldsymbol{\mu})$ such that

$$\forall \boldsymbol{\mu} \in \mathcal{G}, \alpha_{sta}(\boldsymbol{\mu}) \geq \alpha_{LB}(\boldsymbol{\mu})$$

$\alpha_{LB}(\boldsymbol{\mu})$

$$\alpha_{sta}(\boldsymbol{\mu}) = \inf_{v_h \in V_h, v_h \neq 0} \frac{\|\mathbf{A}(\boldsymbol{\mu})\mathbf{v_h}\|_{V'}}{\|\mathbf{v_h}\|_V}.$$

$$\alpha_{sta}^2 = \inf_{v_h \in V_h, v_h \neq 0} \frac{\mathbf{v_h}^T \mathbf{A}(\boldsymbol{\mu})^T \mathbf{M}_V^{-1} \mathbf{A}(\boldsymbol{\mu})\mathbf{v_h}}{\mathbf{v_h}^T \mathbf{M_V} \mathbf{v_h}}$$

$\alpha_{sta}(\boldsymbol{\mu})$ is the square root of the smallest eigenvalue of the problem

$$\mathbf{A}^T(\boldsymbol{\mu})\mathbf{M_V}^{-1}\mathbf{A}(\boldsymbol{\mu})\mathbf{v_h} = \lambda \mathbf{M_V}\ \mathbf{v_h}$$

Aposteriori

$$\mathbf{A}^T \mathbf{M_V}^{-1} \mathbf{A} \mathbf{v_h} = \lambda \mathbf{M_V}\ \mathbf{v_h}$$

eigenvalue of size $\mathcal{N}$ but one can find a lower bound of α_{sta} efficiently computable.

$$(\mathrm{A}(\mathrm{x}, \boldsymbol{\mu}) \nabla u, \nabla u) >= \mu_{min} (\nabla u, \nabla u)$$

Aposteriori

We are going to use

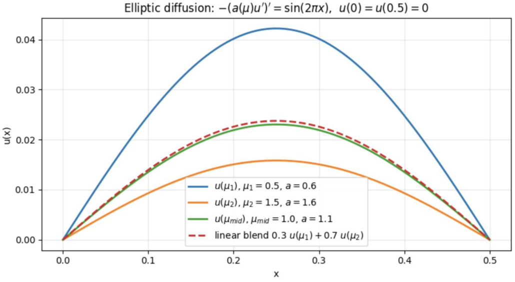
$$\Delta_N(\boldsymbol{\mu}) = \frac{1}{\alpha_{sta}(\boldsymbol{\mu})} \|r_N(\boldsymbol{\mu})\|_{V'} = \frac{1}{\alpha_{sta}(\boldsymbol{\mu})} \|\mathbf{l} - \mathbf{A}\mathbf{u}_\mathbf{N}\|_{V'}$$

An *a posteriori error estimator* is a function $\Delta_N : \mathcal{G} \to \mathbb{R}^+$ satisfying the following properties:

- **Robustness:** ☒ $\forall \boldsymbol{\mu} \in \mathcal{G}, \qquad \|e_N(\boldsymbol{\mu})\|_V \leq \Delta_N(\boldsymbol{\mu})$.
- **Efficiency:** ☒ Since $\alpha\Delta \leq e_N \gamma$, we take $K = \frac{\gamma}{\alpha}$
- **Asymptotic exactness:** ☒ the effectivity index $I_{eff} = \frac{\Delta_N(\boldsymbol{\mu})}{\|u_h - u_N\|} \underset{N\to\infty}{\longrightarrow} 1$
- **Guaranteed upper bound:** ☒ Without evaluating $u_h(\boldsymbol{\mu})$
- **Small evaluation cost:** ☒

TP Aposteriori

TP3: Kolmogorov very small



TP4: same PDE with a posteriori

Greedy algorithm

[1] P. Binev, A. Cohen, W. Dahmen, R. DeVore, G. Petrova, P. Wojtaszczyk, *Convergence rates for greedy algorithms in reduced basis methods.* 2011.

POD vs Greedy

- POD = Smallest mean error

$$\min_{|\Phi_i|=1} \mathbb{E}[\|u - \sum_{i=1}^{N} (u, \Phi_i)\Phi_i\|^2] = \mathbb{E}[\|u - P_{V^N}(u)\|^2]$$

- Greedy $\sim$ Minimizes uniform error

$$u^i = \mathrm{argmax}_{u \in \mathcal{M}} \|u - P_{V^{i-1}}(u)\|,$$

$$V^i = \mathrm{Span}(V^{i-1}, u^i).$$

Greedy minimizes the **worst-case projection error!**

POD vs Greedy

- Kolmogorov N-width:

$$d_N(\mathcal{M}_h, V_h) = \inf_{\substack{V^N \subset V_h \\ \dim(V^N)=N}} \sup_{u_h \in \mathcal{M}_h} \|u_h - P_{V^N}(u)\|_{V_h}$$

- Greedy $\sim$ Smallest uniform error

$$u_h^i = \mathrm{argmax}_{u_h \in \mathcal{M}_h} \|u_h - P_{V_h^{i-1}}(u_h)\|,$$

$$V_h^i = \mathrm{Span}(V_h^{i-1}, u_h^i).$$

Greedy minimizes the **worst-case projection error**

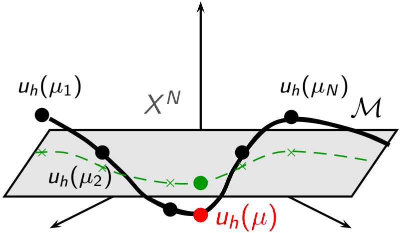
Classical greedy algorithm

$$\mu_0 = \arg\max_{\mu \in \mathcal{G}} \|u_h(\mu_0)\|$$

for $k = 1, \ldots, N$:

$$\mu_k = \arg\max_{\mu \in \mathcal{G}} \|u_h(\mu) - P_{V^{k-1}}(u_h(\mu))\|$$

$P_{V^{k-1}}$:= Projection onto previous RB



Solution manifold

Orthonormal RB in $L^2 := \Phi_j^h, \; j = 1, \ldots, N$

Classical greedy algorithm

Greedy convergence

V^N Greedy RB

$$\sigma_N(\mathcal{M}_h) := \sup_{u_h \in \mathcal{M}_h} \|u_h - P_{V^N}(u)\|_{V_h}$$

Of course $d_N(\mathcal{M}_h, V_h) \leq \sigma_N(\mathcal{M}_h)$

But σ_N not so far from d_N ...

Classical greedy algorithm

Theorem (convergence)

Assume that the Kolmogorov N-width of $\mathcal{M}$ decays exponentially, i.e.

$$\exists \tau, C > 0, \qquad \forall N \geq 1, \qquad d_N(\mathcal{M}, V) \leq Ce^{-\tau N}.$$

Assume moreover that the decay rate satisfies $\tau > \log 2$. Then there exist $\tau', C' > 0$ such that the sequence of subspaces $(V^n)_{n \geq 1}$ generated by the (strong) greedy algorithm satisfies

$$\forall N \geq 1, \qquad \forall u \in \mathcal{M}, \qquad \|u - P_{V_N}(u)\|_V \leq C' e^{-\tau' N}.$$

Classical greedy algorithm

Theorem (convergence)

Assume that the Kolmogorov N-width of $\mathcal{M}$ decays exponentially, i.e.

$$\exists \tau, C > 0, \qquad \forall N \geq 1, \qquad d_N(\mathcal{M}, V) \leq Ce^{-\tau N}.$$

Assume moreover that the decay rate satisfies $\tau > \log 2$.

$$\sigma_N \lesssim 2^N d_N(\mathcal{M}, V)$$

Then there exist $\tau', C' > 0$ such that the sequence of subspaces $(V^n)_{n \geq 1}$ generated by the (strong) greedy algorithm satisfies

$$\forall N \geq 1, \qquad \forall u \in \mathcal{M}, \qquad \|u - P_{V_N}(u)\|_V \leq C' e^{-\tau' N}.$$

Classical greedy algorithm

With our model problem: $u(\boldsymbol{\mu}) \in V$ is the solution of

$$\forall v \in V, \qquad a(u(\boldsymbol{\mu}), v; \boldsymbol{\mu}) = \ell(v; \boldsymbol{\mu}),$$

The Galerkin reduced basis approximation $u_N(\mu) \in V_N$ is solution of

$$\forall v_N \in V_N, \qquad a(u_N(\boldsymbol{\mu}), v_N; \boldsymbol{\mu}) = \ell(v_N; \boldsymbol{\mu}).$$

Cea's lemma

$$\|u(\mu) - u_N(\mu)\|_V \leq \frac{\gamma(\mu)}{\alpha(\mu)} \|u(\mu) - P_{V_N}(u(\mu))\|_V.$$

Therefore, $\|u(\mu) - u_N(\mu)\|_V \longrightarrow 0 \qquad$ as $N \to +\infty$, exponentially.

Discretization: Strong greedy algorithm

Set $\mu_0 = \arg\max_{\mu \in \mathcal{G}} \|u_h(\mu_0)\|$

Normalize $u_h^0 = u_h(\mu_0)$

For $k = 1, \ldots, N$:

  Set $\mu_k = \arg\max_{\mu \in \mathcal{G}} \|u_h(\mu) - P_{V^{k-1}}(u_h(\mu))\|$

  Orthonormalize $u_h^k = u_h(\mu_k)$ with a Gram-schmidt process

  $V^k = V^{k-1} \oplus \mathrm{Span}\{u_h(\mu^k)\}$

How do we choose N?

Stop when $\arg\max_{\mu \in \mathcal{G}} \|u_h(\mu) - P_{V^{k-1}}(u_h(\mu))\| < \varepsilon$

Discretization: Strong greedy algorithm

In practice:

1: Set $\mu_0 = \underset{\mu \in \mathcal{G}_{train}}{\arg\max} \|u_h(\mu_0)\|$
2: Normalize $u_h^0 = u_h(\mu_0)$
3: For $k = 1, \ldots, N$:
4: Set $\mu_k = \underset{\mu \in \mathcal{G}_{train}}{\arg\max} \|u_h(\mu) - P_{V^{k-1}}(u_h(\mu))\|$
5: Orthonormalize $u_h^k = u_h(\mu_k)$ with a Gram-schmidt process
6: $V^k = V^{k-1} \oplus \mathrm{Span}\{u_h(\mu^k)\}$
7: Stop when $\underset{\mu \in \mathcal{G}_{train}}{\arg\max} \|u_h(\mu) - P_{V^{k-1}}(u_h(\mu))\| < \varepsilon$

Discretization: Weak greedy algorithm

In practice: $\|u_h(\mu) - P_{V^{k-1}}(u_h(\mu))\| \leq \|u_h(\mu) - u_N\| \leq \Delta_N(\mu)$

1: Set $\mu_0 = \underset{\mu \in \mathcal{G}_{train}}{\arg\max} \|u_h(\mu_0)\|$
2: Normalize $u_h^0 = u_h(\mu_0)$
3: For $k = 1, \ldots, N$:
4: Set $\mu_k = \underset{\mu \in \mathcal{G}_{train}}{\arg\max}\ \Delta_k(\mu)$
5: Orthonormalize $u_h^k = u_h(\mu_k)$ with a Gram-schmidt process
6: $V^k = V^{k-1} \oplus \mathrm{Span}\{u_h(\mu^k)\}$
7: Stop when $\underset{\mu \in \mathcal{G}_{train}}{\arg\max} \|u_h(\mu) - P_{V^{k-1}}(u_h(\mu))\| < \varepsilon$

Discretization: Weak greedy algorithm

In practice: Exhaustive search over the discrete sample set $\mathcal{G}_{train}$

- ◇ The complexity of one iteration is linear in $\mathrm{Card}(\mathcal{G}_{train})$, with
  - RB assembly: $\mathcal{O}\left(N^2(Q^a)^2 + NQ^\ell\right)$,
  - RB solve: $\mathcal{O}(N^3)$,
  - RB residual norm: $\mathcal{O}\left(N^2(Q^a)^2 + NQ^aQ^\ell + (Q^\ell)^2\right)$.

  Hence, the total cost is

$$\mathcal{O}\left(\mathrm{Card}(\mathcal{G}_{train})\left(N^3 + 2N^2(Q^a)^2 + NQ^\ell Q^a + (Q^\ell)^2\right)\right).$$

- ◇ Typical choices for $\mathcal{G}_{train}$: Cartesian grid, random samples, ...
- ◇ Instead of using the same training set $\mathcal{G}_{train}$ at every Greedy iteration, one may change $\mathcal{G}_{train}$ from one iteration to the next (resampling).

Conclusion: Two algorithms

We now have two methods for constructing a reduced basis:

- **POD approach:** N_{train} ($> N$) high-fidelity solutions are computed to build a basis of size N
  - ✓ The exploration/compression paradigm is very easy to implement,
  - × The cost can be high if one wants to avoid missing relevant information
- **Greedy approach:** N high-fidelity solutions are computed to build a basis of size N
  - × More demanding to implement (incremental construction of the basis),
  - ✓ Lower cost to avoid missing relevant information (the number of HF solves is independent of the cardinality of $\mathcal{G}_{train}$).

TP Aposteriori

TP4 bis: Greedy + Aposteriori