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

**Elise Grosjean**
Ensta-Paris

Ensta-Paris
Institut Polytechnique de Paris

# Reduced Basis Methods (RBM)

## Solution manifold

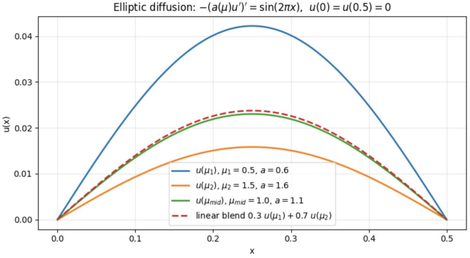
Solution manifold:
$\mathcal{M} = \{u(\boldsymbol{\mu})|\ \boldsymbol{\mu} \in \mathcal{G}\}$
HF solution manifold:
$\mathcal{M}_h = \{u_h(\boldsymbol{\mu})|\ \boldsymbol{\mu} \in \mathcal{G}\}$

**How can we reduce the manifold complexity?**

- ◇ Keep HF precision
- ◇ Reduce computational costs

## Reduced space

$V^N$ Reduced space



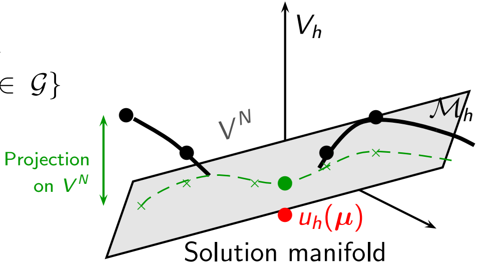

$$V_N = \mathrm{Span}\{u_1, u_2, \ldots u_N\}$$

where $u_1, \ldots, u_N =$ snapshots

# Reduced Basis Methods (RBM)

## Solution manifold

Solution manifold: $\mathcal{M} = \{u(\boldsymbol{\mu}) |\ \boldsymbol{\mu} \in \mathcal{G}\}$
HF solution manifold: $\mathcal{M}_h = \{u_h(\boldsymbol{\mu}) |\ \mu \in\ \mathcal{G}\}$




## Reduced space

$V^N$ Reduced space

Projection on $V^N$:

$$\boxed{\inf_{v_N \in V^N} \|u_h - v_N\|_{V_h}.}$$

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

$$\boxed{?}$$

Exponential decay

$$\boxed{\forall N > 1, \quad d_N(\mathcal{M}, V) \leq Ce^{-\tau N}.}$$

# TP RB+VF

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

# TP RB+VF

- **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+VF

◇ **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+VF

- **Offline** Construction of a reduced space $V_N$ spanned by a reduced basis $\rightarrow$ ???
  $V_M = Span\{u_1, \ldots, u_M\}$ (or $V_{Ntrain} = Span\{u_1, \ldots, u_{Ntrain}\}$)
  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+VF

- **Offline** Construction of a reduced space $V_N$ spanned by a reduced basis $\rightarrow$ ???
  $V_M = Span\{u_1, \ldots, u_M\}$ (or $V_{Ntrain} = Span\{u_1, \ldots, u_{Ntrain}\}$)
  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_tpfa_rom(mu, Nx, Ny, Phi)
```

# TP RB+VF

`assemble_tpfa`.

## Finite Volume Methods

Based on the conservation form of the PDE $\rightarrow$ Flux: total outward flux $=$ the total internal source

Integrate the equation on each cell $\kappa$ and apply Stokes' formula:

$$\int_K f(\mathbf{x})\ d(\mathbf{x}) = -\int_K \nabla \cdot \big(a(\boldsymbol{\mu})\nabla u\big) = \sum_{\sigma\in\mathcal{F}_K} \underbrace{-\int_\sigma a(\boldsymbol{\mu})\nabla u(\mathbf{x}) \cdot \mathbf{n}_{K,\sigma}\ d_\gamma(\mathbf{x})}_{\overline{F}_{K,\sigma}}$$

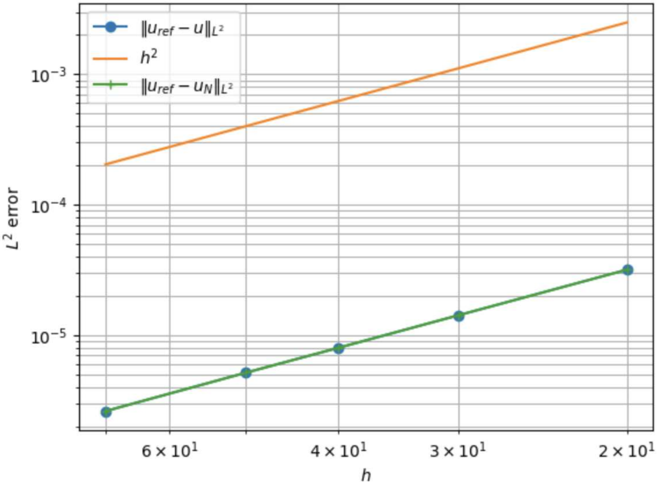
Approximate each flux and write the discrete balance equation obtained.

$$\Longrightarrow \overline{F}_{K,\sigma} = \underbrace{-\overline{A(\boldsymbol{\mu})}|\sigma|\frac{u(x_L)-u(x_K)}{d_{KL}}}_{F_{K,\sigma}(u_h)} + \mathcal{O}(h^2)$$

where $\overline{A}$ is the harmonic average: $\overline{A} = \frac{A(x_L)A(x_K)d_{KL}}{A(x_L)d_{K,\sigma}+A(x_K)d_{L,\sigma}}$

# TP RB+VF

Expected results: Convergence in $\mathcal{O}(h^2)$
$\mu = (0.6, 0.5, 0.2, 0.8)$



# TP RB+VF

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

# TP RB+VF

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

# TP RB+VF

```
def A_fct(x, y, mu1, mu2):
 # Diffusion parameter
 return 2.0 * mu1 + mu2 * np.sin(x + y) * np.cos(x * y)

def f_fct(x, y, mu3, mu4):
 # Right-hand side term
return mu3 * (1.0 - y) + mu4 * x * (1.0 - x)
```

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

$\overline{A}$ is the harmonic average: $\overline{A} = \frac{A(x_L)A(x_K)d_{KL}}{A(x_L)d_{K,\sigma}+A(x_K)d_{L,\sigma}}$

# TP RB+VF

```
def A_fct(x, y, mu1, mu2):
 # Diffusion parameter
 return 2.0 * mu1 + mu2 * np.sin(x + y) * np.cos(x * y)

def f_fct(x, y, mu3, mu4):
 # Right-hand side term
return mu3 * (1.0 - y) + mu4 * x * (1.0 - x)
```

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

$\overline{A}$ is the harmonic average: $\overline{A} = \frac{A(x_L)A(x_K)d_{KL}}{A(x_L)d_{K,\sigma}+A(x_K)d_{L,\sigma}}$

# TP RB+VF

Find $u = \sum_i a_i(\boldsymbol{\mu})\Phi_i(x)$

```
# assemble full system
_,_,M, b = assemble_tpfa(Nx=Nx, Ny=Ny, mu=mu)

def solve_tpfa_rom(mu, Nx, Ny, Phi):
 # Reduced operators
 Mr =(Phi.T  (M  Phi))
 br = (Phi.T  b)
 a = np.linalg.solve(Mr, br)
 u_rom = np.dot(Phi, a)
 U_rom = u_rom.reshape((Nx, Ny), order="F")
 return a, U_rom
```

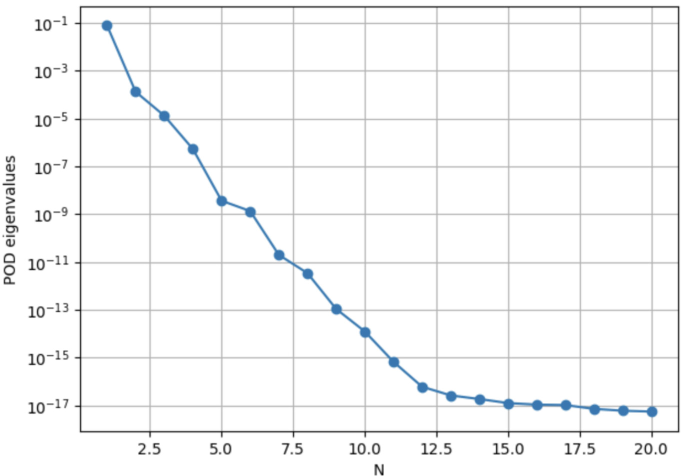
# TP RB+VF

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

# TP RB+VF

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

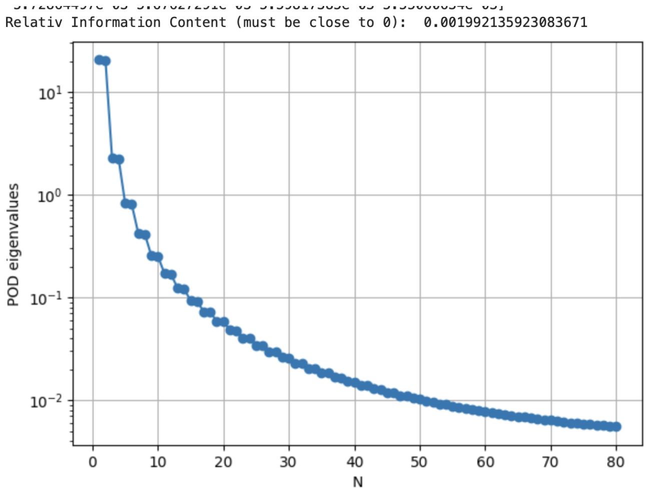
```
Relativ Information Content (must be close to 0):  -1.7763568394002505e-15
```



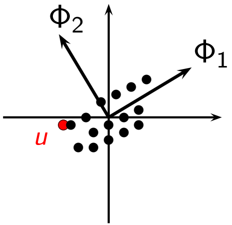
# TP Kolmogorov

TP2:

with Kolmogorov $n$ width not small: $u(x,\mu) = tanh(\frac{x-\mu}{\delta})$.



# Computable error bound

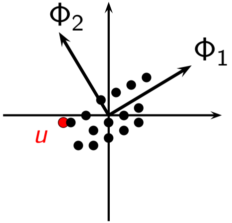



$$\boxed{d_N(\mathcal{M}_h, V_h) = \inf_{\substack{V^N \subset V_h \\ \dim(V^N)=N}} \sup_{u_h \in \mathcal{M}_h} \inf_{v_N \in V^N} \|u_h - v_N\|_{V_h} = \varepsilon.}$$

**POD reminder:** $V_N = \mathrm{Span}\{\Phi_1, \ldots, \Phi_N\}$, where

$$V_N = \inf_{\substack{V^N \subset V^M \\ \dim(V^N)=N}} \mathbb{E}[\|u - P_N u\|_V^2]$$

# Computable error bound



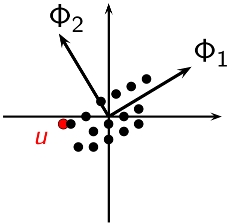

$$\boxed{d_N(\mathcal{M}_h, V_h) = \inf_{\substack{V^N \subset V_h \\ \dim(V^N)=N}} \sup_{u_h \in \mathcal{M}_h} \inf_{v_N \in V^N} \|u_h - v_N\|_{V_h} = \varepsilon.}$$

**POD reminder:** $V_N = \mathrm{Span}\{\Phi_1, \ldots, \Phi_N\}$, where

$$V_N = \inf_{\substack{V^N \subset V^M \\ \dim(V^N)=N}} \mathbb{E}[\|u - P_N u\|_V^2]$$

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

# Computable error bound



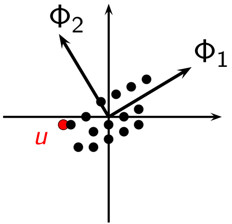

$$\boxed{d_N(\mathcal{M}_h, V_h) = \inf_{\substack{V^N \subset V_h \\ \dim(V^N)=N}} \sup_{u_h \in \mathcal{M}_h} \inf_{v_N \in V^N} \|u_h - v_N\|_{V_h} = \varepsilon.}$$

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

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

Thus, at the points $\{\mu_m\}_{1 \leq m \leq M}$, we know that the error is bounded by $\delta > 0$

# Computable error bound



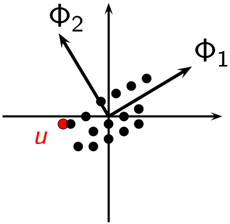

$$d_N(\mathcal{M}_h, V_h) = \inf_{\substack{V^N \subset V_h \\ \dim(V^N)=N}} \sup_{u_h \in \mathcal{M}_h} \inf_{v_N \in V^N} \|u_h - v_N\|_{V_h} = \varepsilon.$$

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 $\mu$, 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



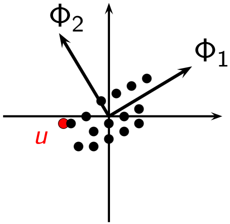

$$\boxed{d_N(\mathcal{M}_h, V_h) = \inf_{\substack{V^N \subset V_h \\ \dim(V^N)=N}} \sup_{u_h \in \mathcal{M}_h} \inf_{v_N \in V^N} \|u_h - v_N\|_{V_h} = \varepsilon.}$$

Like FEM Cea's lemma, under some assumptions ($a$ coercive ...)

$$\|u_h - u_N\|_V \leq \frac{\gamma(\mu)}{\alpha(\mu)} \mathit{inf}_{v_N \in V_N} \|u_h - v_N\|_V$$

but ...

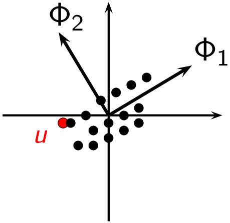
# Computable error bound




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

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

# Computable error bound




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

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

**aposteriori bound !!!**

# Aposteriori

The aposteriori error bound measures the RB method fiability:

$$\|u(\boldsymbol{\mu}) - u_N(\boldsymbol{\mu})\|_V \leq \underbrace{\|u(\boldsymbol{\mu}) - u_h(\boldsymbol{\mu})\|_V}_{=\mathcal{O}(h^s)} + \underbrace{\|u_h(\boldsymbol{\mu}) - u_N(\boldsymbol{\mu})\|_V}_{\leq \Delta_N(\boldsymbol{\mu})}$$

There are many different types of a posteriori error estimation:

- residual-based estimates
- averaging-based estimates
- equilibrated fluxes estimates
- equilibrated residual estimates
- hierarchical estimates
- heuristic estimates

# Computable error bound

## Theorem: Banach-Necas-Babuska

Let $a : V \times W$ continuous bilinear form and $f : W \to \mathbb{R}$ continuous linear form
The problem

$$\text{find } u \in V,\ a(u,w) = f(w),\ \forall w \in W$$

is well-posed iff

$$\exists \alpha_{sta} > 0, \forall v \in V,\ \sup_{w \in W \setminus \{0\}} \frac{a(v,w)}{\|w\|_W} \geq \alpha_{sta} \|v\|_V.$$

The following estimate holds true:

$$\boxed{\|u\|_V \leq \frac{1}{\alpha_{sta}} \|f\|_{W'}}$$

The dual space of $V$ is $V' := \mathcal{L}(V, \mathbb{R})$ the space of continuous linear forms.
Norm: $\|g\|_{V'} = \sup_{v \in V, v \neq 0} \frac{|g(v)|}{\|v\|_V}$

# Computable error bound

## Theorem: Banach-Necas-Babuska

Let $a : V_h \times V_h$ continuous bilinear form and $f_h : V_h \to \mathbb{R}$ continuous linear form
The H-F problem

$$\text{find } u_h \in V_h,\ a(u_h, w_h) = f(w_h),\ \forall w_h \in V_h$$

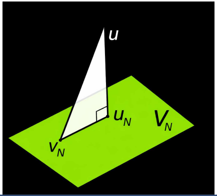
is well-posed iff

$$\exists \alpha_{sta} > 0, \forall v_h \in V_h, \sup_{w_h \in V_h \setminus \{0\}} \frac{a(v_h, w_h)}{\|w_h\|_V} \geq \alpha_{sta} \|v_h\|_V.$$

The following estimate holds true:

$$\boxed{\|u_h\|_V \leq \frac{1}{\alpha_{sta}} \|f_h\|_{V'}}$$

# Computable error bound




## Reminder: Galerkin condition

The error $e_N$ is orthogonal to the test space $V_N$ in the energy inner product induced by $a(\cdot,\cdot)$ :

$$\boxed{a(e_N, v_N) = a(u - u_N, v_N) = 0, \ \forall v_N \in V_N.}$$

Consider the weak residual $r(v) = \ell(v) - a(u_N, v)$. Then

$$r(v_N) = a(u, v_N) - a(u_N, v_N) = a(e_N, v_N) = 0, \ \forall v_N \in V_N.$$

In other words, the residual vanishes on the test space.

# Computable error bound

The following estimate holds true:

$$\boxed{\|u_h\|_V \leq \frac{1}{\alpha_{sta}} \|f_h\|_{V'}}$$

And thus:
the following estimate holds true:

$$\|u_h - u_N\| = \|e_h\|_V \leq \frac{1}{\alpha_{sta}} \|r_h\|_{V'}$$

# Computable error bound

$\alpha_{sta} = \inf_{v_h \in V_h} \sup_{w_h \in V_h} \frac{a(v_h, w_h)}{\|v_h\|_V \|w_h\|_V} > 0$ is the "inf-sup stability constant"

$$\mathbf{A} : V \to V'$$

With HF basis $\{w_i\}_{i=1}^{\mathcal{N}}$, we know that:
$a(u, v) = \mathbf{v}^T \mathbf{A} \mathbf{u}$, with $\mathbf{A}_{ij} = a(w_j, w_i)$, and $(u, v) = \mathbf{v}^T \mathbf{M} \mathbf{u}$ with $\mathbf{M}_{ij} = (w_j, w_i)$.

$\alpha_{sta} = \inf_{v_h \in V_h} \sup_{w_h \in V_h} \frac{\mathbf{w_h}^T \mathbf{A} \ \mathbf{v_h}}{\|\mathbf{v_h}\|_V \|\mathbf{w_h}\|_V} = \inf_{v_h \in V_h} \frac{\|\mathbf{A} \mathbf{v_h}\|_{V'}}{\|\mathbf{v_h}\|_V} > 0$ is the "inf-sup stability constant"

$$\forall u_h \in V_h, \ \|\mathbf{A} \mathbf{u_h}\|_{V'} \geq \alpha_{sta} \|\mathbf{u_h}\|_V$$

# Computable error bound

$\alpha_{sta} = \inf_{v_h \in V_h} \sup_{w_h \in V_h} \frac{a(v_h, w_h)}{\|v_h\|_V \|w_h\|_V} > 0$ is the "inf-sup stability constant"

$$\mathbf{A} : V \to V'$$

With HF basis $\{w_i\}_{i=1}^{\mathcal{N}}$, we know that:
$a(u, v) = \mathbf{v}^T \mathbf{A} \mathbf{u}$, with $\mathbf{A}_{ij} = a(w_j, w_i)$, and $(u, v) = \mathbf{v}^T \mathbf{M} \mathbf{u}$ with $\mathbf{M}_{ij} = (w_j, w_i)$.

$\alpha_{sta} = \inf_{v_h \in V_h} \sup_{w_h \in V_h} \frac{\mathbf{w_h}^T \mathbf{A} \ \mathbf{v_h}}{\|\mathbf{v_h}\|_V \|\mathbf{w_h}\|_V} = \inf_{v_h \in V_h} \frac{\|\mathbf{A} \mathbf{v_h}\|_{V'}}{\|\mathbf{v_h}\|_V} > 0$ is the "inf-sup stability constant"

$$\forall u_h \in V_h, \ \|\mathbf{A} \mathbf{u_h}\|_{V'} \geq \alpha_{sta} \|\mathbf{u_h}\|_V$$

# Computable error bound

$\alpha_{sta} = \inf_{v_h \in V_h} \sup_{w_h \in V_h} \frac{a(v_h, w_h)}{\|v_h\|_V \|w_h\|_V} > 0$ is the "inf-sup stability constant"

$$\mathbf{A} : V \to V'$$

With HF basis $\{w_i\}_{i=1}^{\mathcal{N}}$, we know that:
$a(u, v) = \mathbf{v}^T \mathbf{A} \mathbf{u}$, with $\mathbf{A}_{ij} = a(w_j, w_i)$, and $(u, v) = \mathbf{v}^T \mathbf{M} \mathbf{u}$ with $\mathbf{M}_{ij} = (w_j, w_i)$.

$\alpha_{sta} = \inf_{v_h \in V_h} \sup_{w_h \in V_h} \frac{\mathbf{w_h}^T \mathbf{A} \ \mathbf{v_h}}{\|\mathbf{v_h}\|_V \|\mathbf{w_h}\|_V} = \inf_{v_h \in V_h} \frac{\|\mathbf{A}\mathbf{v_h}\|_{V'}}{\|\mathbf{v_h}\|_V} > 0$ is the "inf-sup stability constant"

$$\forall u_h \in V_h, \ \|\mathbf{A}\mathbf{u_h}\|_{V'} \geq \alpha_{sta} \|\mathbf{u_h}\|_V$$

# Computable error bound

$\alpha_{sta} = \inf_{v_h \in V_h} \sup_{w_h \in V_h} \frac{a(v_h, w_h)}{\|v_h\|_V \|w_h\|_V} > 0$ is the "inf-sup stability constant"

$$\mathbf{A} : V \to V'$$

With HF basis $\{w_i\}_{i=1}^{\mathcal{N}}$, we know that:
$a(u, v) = \mathbf{v}^T \mathbf{A} \mathbf{u}$, with $\mathbf{A}_{ij} = a(w_j, w_i)$, and $(u, v) = \mathbf{v}^T \mathbf{M} \mathbf{u}$ with $\mathbf{M}_{ij} = (w_j, w_i)$.

$\alpha_{sta} = \inf_{v_h \in V_h} \sup_{w_h \in V_h} \frac{\mathbf{w_h}^T \mathbf{A} \ \mathbf{v_h}}{\|\mathbf{v_h}\|_V \|\mathbf{w_h}\|_V} = \inf_{v_h \in V_h} \frac{\|\mathbf{A} \mathbf{v_h}\|_{V'}}{\|\mathbf{v_h}\|_V} > 0$ is the "inf-sup stability constant"

$$\forall u_h \in V_h, \ \|\mathbf{A} \mathbf{u_h}\|_{V'} \geq \alpha_{sta} \|\mathbf{u_h}\|_V$$

# Computable error bound

$$\forall u_h \in V_h,\ \|\mathbf{Au_h}\|_{V'} \geq \alpha_{sta}\|\mathbf{u_h}\|_V$$

Coercivity?

- $a$ can be asymetric
- $a(u,u)$ can be 0

Stokes:

$$(u\ p)\begin{pmatrix} A & B^T \\ B & 0 \end{pmatrix}\begin{pmatrix} u \\ p \end{pmatrix} = u^T A u + \underbrace{2p^T B u}_{\text{can be negative!}}$$

Coercivity $\Leftrightarrow u = \min_{v\in V} J(v) = \frac{1}{2}a(v,v) - l(v)$ ( Convexity, unique minimum: Lax-Milgram)

# Computable error bound

$$\forall u_h \in V_h, \ \|\mathbf{Au_h}\|_{V'} \geq \alpha_{sta}\|\mathbf{u_h}\|_V$$

Coercivity?

- $a$ can be asymetric
- $a(u,u)$ can be 0

Stokes:

$$\begin{pmatrix} u & p \end{pmatrix} \begin{pmatrix} A & B^T \\ B & 0 \end{pmatrix} \begin{pmatrix} u \\ p \end{pmatrix} = u^T A u + \underbrace{2p^T B u}_{\text{can be negative!}}$$

Coercivity $\Leftrightarrow u = \min_{v \in V} J(v) = \frac{1}{2}a(v,v) - l(v)$ ( Convexity, unique minimum: Lax-Milgram)

# Computable error bound

$$\forall u_h \in V_h, \ \|\mathbf{Au_h}\|_{V'} \geq \alpha_{sta}\|\mathbf{u_h}\|_V$$

Going back to $V_N \subset V_h$:

consider the weak residual $r(v) = \ell(v) - a(u_N, v)$. Then

$$\begin{aligned} V'_N \ni \mathbf{r_N} &= \mathbf{l} - \mathbf{Au_N} \\ &= \mathbf{Au_h} - \mathbf{Au_N} = \mathbf{Ae_N} \end{aligned}$$

- From continuity, $\|\mathbf{Aw}\|_{V'} \leq \gamma\|\mathbf{w}\|_V$
- From inf-sup condition, $\|\mathbf{Aw}\|_{V'} \geq \alpha_{sta}\|\mathbf{w}\|_V$

Thus, with $\mathbf{w} = \mathbf{e_N}$,

$$\boxed{\frac{1}{\gamma}\|\mathbf{r_N}\|_{W'} \leq \|\mathbf{e_N}\|_V \leq \frac{1}{\alpha_{sta}}\|\mathbf{r_N}\|_{V'}}$$

# Computable error bound

$$\forall u_h \in V_h,\ \|\mathbf{Au_h}\|_{V'} \geq \alpha_{sta}\|\mathbf{u_h}\|_V$$

Going back to $V_N \subset V_h$:
consider the weak residual $r(v) = \ell(v) - a(u_N, v)$. Then

$$\begin{aligned} V_N' \ni \mathbf{r_N} &= \mathbf{l} - \mathbf{Au_N} \\ &= \mathbf{Au_h} - \mathbf{Au_N} = \mathbf{Ae_N} \end{aligned}$$

- ◇ From continuity, $\|\mathbf{Aw}\|_{V'} \leq \gamma\|\mathbf{w}\|_V$
- ◇ From inf-sup condition, $\|\mathbf{Aw}\|_{V'} \geq \alpha_{sta}\|\mathbf{w}\|_V$

Thus, with $\mathbf{w} = \mathbf{e_N}$,

$$\boxed{\frac{1}{\gamma}\|\mathbf{r_N}\|_{W'} \leq \|\mathbf{e_N}\|_V \leq \frac{1}{\alpha_{sta}}\|\mathbf{r_N}\|_{V'}}$$

# 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 (and local 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 $I_{eff} = \frac{\Delta_N(\boldsymbol{\mu})}{\|u_h - u_N\|} \underset{N \to \infty}{\longrightarrow} 1$

**Guaranteed upper bound:** The function $\Delta_N$ 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

We are going to use

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

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

**Robustness:** $\boxtimes$ $\forall \boldsymbol{\mu} \in \mathcal{G}, \qquad \|e_N(\boldsymbol{\mu})\|_V \leq \Delta_N(\boldsymbol{\mu})$.

**Efficiency (and local efficiency):** $\boxtimes$ Since $\alpha\Delta \leq e_N\gamma$, we take $K = \frac{\gamma}{\alpha}$

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

**Guaranteed upper bound:** $\boxtimes$ Without evaluating $u_h(\boldsymbol{\mu})$

**Small evaluation cost:** local residual (useful for raffinement / mesh adaptation)

**Error components identification:** Distinguish and estimate separately the different error components (e.g. algebraic error, linked to solver imprecision)

# Aposteriori

We are going to use

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

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 (and local efficiency):** ☒ Since $\alpha\Delta \leq e_N\gamma$, we take $K = \frac{\gamma}{\alpha}$

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

**Guaranteed upper bound:** ☒ Without evaluating $u_h(\boldsymbol{\mu})$

**Small evaluation cost:** local residual (useful for raffinement / mesh adaptation)

**Error components identification:** Distinguish and estimate separately the different error components (e.g. algebraic error, linked to solver imprecision)

# Aposteriori

We are going to use

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

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 (and local 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:** local residual (useful for raffinement / mesh adaptation)

**Error components identification:** Distinguish and estimate separately the different error components (e.g. algebraic error, linked to solver imprecision)

# Aposteriori

We are going to use

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

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

**Robustness:** $\boxtimes$ $\forall \boldsymbol{\mu} \in \mathcal{G}, \qquad \|e_N(\boldsymbol{\mu})\|_V \leq \Delta_N(\boldsymbol{\mu})$.

**Efficiency (and local efficiency):** $\boxtimes$ Since $\alpha\Delta \leq e_N\gamma$, we take $K = \frac{\gamma}{\alpha}$

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

**Guaranteed upper bound:** $\boxtimes$ Without evaluating $u_h(\boldsymbol{\mu})$

**Small evaluation cost:** local residual (useful for raffinement / mesh adaptation)

**Error components identification:** Distinguish and estimate separately the different error components (e.g. algebraic error, linked to solver imprecision)

# Aposteriori

We are going to use

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

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

**Robustness:** $\boxtimes$ $\forall \boldsymbol{\mu} \in \mathcal{G}, \qquad \|e_N(\boldsymbol{\mu})\|_V \leq \Delta_N(\boldsymbol{\mu}).$

**Efficiency (and local efficiency):** $\boxtimes$ Since $\alpha\Delta \leq e_N\gamma$, we take $K = \frac{\gamma}{\alpha}$

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

**Guaranteed upper bound:** $\boxtimes$ Without evaluating $u_h(\boldsymbol{\mu})$

**Small evaluation cost:** ??? local residual (useful for raffinement / mesh adaptation)

**Error components identification:** Distinguish and estimate separately the different error components (e.g. algebraic error, linked to solver imprecision)

# Aposteriori

We are going to use

$$\boxed{\Delta_N(\boldsymbol{\mu}) = \frac{1}{\alpha_{sta}(\boldsymbol{\mu})}\|r(u_N(\boldsymbol{\mu}))\|_{V'} = \frac{1}{\alpha_{sta}(\boldsymbol{\mu})}\|\mathbf{l} - \mathbf{A}\mathbf{u_N}\|_{V'}}$$

The aposteriori error bound measures the RB method fiability:

$$\|u(\boldsymbol{\mu}) - u_N(\boldsymbol{\mu})\|_V \leq \underbrace{\|u(\boldsymbol{\mu}) - u_h(\boldsymbol{\mu})\|_V}_{=\mathcal{O}(h^s)} + \underbrace{\|u_h(\boldsymbol{\mu}) - u_N(\boldsymbol{\mu})\|_V}_{\leq \Delta_N(\boldsymbol{\mu})}$$

There are many different types of a posteriori error estimation:

- residual-based estimates
- averaging-based estimates
- equilibrated fluxes estimates
- equilibrated residual estimates
- hierarchical estimates
- heuristic estimates

# 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: $\alpha^* = \inf\limits_{v \neq 0} \frac{a(v,v)}{\|v\|_V^2}$

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

## Reminder: Riesz representation Theorem

$$\forall r \in V', \quad \exists! z \in V \text{ such that } r(v) = (z, v)_V \quad \forall v \in V,$$

and the norm on $V'$ is:

$$\|r\|_{V'}^2 = r(z) = (z, z)_V.$$

Thus, since in our setting $V = H_0^1(\Omega)$, thus $(z, z)_V = z^T \mathbf{K} z = r^T \mathbf{K}^{-1} r$, where $\mathbf{K}$ is the stiffness matrix.

# Dual norm of the residual

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

# Reduced basis Galerkin approximation

Assume the weak formulation of the HF problem yields the discretized system

$$\mathbf{A}(\boldsymbol{\mu})\mathbf{u}(\boldsymbol{\mu}) = \mathbf{l}(\boldsymbol{\mu})$$

then

$$a\left(u_N(\boldsymbol{\mu}), v_N; \boldsymbol{\mu}\right) = \ell\left(v_N; \boldsymbol{\mu}\right)$$

gives a new system to solved:

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

# Reduced basis Galerkin approximation

Assembling cost with the affine operators:
$\mathcal{O}(N^2 Q^a + N Q^l)$ with

$$\mathbf{P}^T \mathbf{A}(\boldsymbol{\mu}) \mathbf{P} = \sum_{q=1}^{Q^a} \theta_q^a(\boldsymbol{\mu}) \underbrace{\mathbf{P}^T \mathbf{A}_q \mathbf{P}}_{\text{precomputed offline}} \quad , \quad \mathbf{P}^T \mathbf{l}(\boldsymbol{\mu}) = \sum_{q=1}^{Q^l} \theta_q^l(\boldsymbol{\mu}) \underbrace{\mathbf{P}^T \mathbf{l}_q}_{\text{precomputed offline}} \quad .$$

# Dual norm of the residual

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

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

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

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

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

# Aposteriori

We are going to use

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

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 (and local 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:** ☒

**Error components identification:** Distinguish and estimate separately the different error components (e.g. algebraic error, linked to solver imprecision)

# Aposteriori

Let's get back to our sheep

linear second-order parameter dependent problem

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

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

$$\Delta_N = \frac{1}{\alpha}\sum_K \eta_K$$

where

$$\eta_K^2 = \Big(h_K^2 \|f + \Delta u_N\|^2_{L^2(K)} + \sum_{e \subset \partial K} h_e \|[[\nabla u_N \cdot \mathbf{n}]]\|^2_{L^2(e)}\Big)$$

# Aposteriori

$$\Delta_N = \frac{1}{\alpha}\sum_K \eta_K \text{ where } \eta_K^2 = \left(h_K^2\|f+\Delta u_N\|^2_{L^2(K)} + \sum_e h_e\|[[\nabla u_N \cdot \mathbf{n}]]\|^2_{L^2(e)}\right)$$

**Proof:**
$r(v) = \ell(v) - a(u_N, v)$. Then with $\Omega = \cup_K K$,

$$\begin{aligned} r(v) &= \sum_K \Big( \int_K fv - \int_K (\nabla u_N, \nabla v)\Big) \\ &= \sum_K \Big( \int_K fv + \int_K \Delta u_N \cdot v - \int_{\partial K} (\nabla u_N \cdot n)v\Big) \\ &= \sum_K \underbrace{\int_K (f+\Delta u_N)v}_{\text{element residual}} - \sum_K \underbrace{\int_{\partial K} (\nabla u_N \cdot n)v}_{\text{flux continuity default}} \end{aligned}$$

# Aposteriori

By C-S and Poincare:

$|\int_K (f+\Delta u_N)v| \leq \|f+\Delta u_N\|_{L^2(K)}\|v\|_{L^2(K)} \leq Ch_K\|f+\Delta u_N\|_{L^2(K)}\|\nabla v\|_{L^2(K)}$

and by trace inequality,

$|\int_e (\nabla u_N \cdot n)v| \leq \|\nabla u_N \cdot \mathbf{n}\|_{L^2(e)}\|v\|_{L^2(e)} \leq Ch_e^{1/2}\|\nabla u_N \cdot \mathbf{n}\|_{L^2(e)}\|\nabla v\|_{L^2(K)}$

$$|r(v)| \leq \Big(\sum_K Ch_K\|f+\Delta u_N\|_{L^2(K)} + Ch_e^{1/2}\|\nabla u_N \cdot \mathbf{n}\|_{L^2(e)}\Big)\|\nabla v\|_{L^2(K)}$$

Thus $(A+B \leq \sqrt{2}(A^2+B^2)^{1/2} \Leftrightarrow)$,

$|r(v)| \leq \Big(\sum Ch_K^2\|f+\Delta u_N\|^2_{L^2(K)} + Ch_e\|\nabla u_N \cdot \mathbf{n}\|^2_{L^2(e)}\Big)^{1/2}\|\nabla v\|_{L^2(K)}$

$$\|r\|_{V'} = \sup_{v\neq 0}\frac{|r(v)|}{\|v\|_V}$$

thus

$$\Delta_N = \frac{1}{\alpha}\sum_K \eta_K \ \text{ where } \ \eta_K^2 = \Big(h_K^2\|f+\Delta u_N\|^2_{L^2(K)} + \sum_e h_e\|[[\nabla u_N \cdot \mathbf{n}]]\|^2_{L^2(e)}\Big)$$

# Aposteriori

The error in $K$ depends not only on the residual inside $K$ but also on nearby elements: so the estimator uses a patch residual.

$$\Delta_N = \frac{1}{\alpha_{sta}}(\eta_1 + \eta_2)$$

where

$$\eta_1^2 = \sum_{K' \in I_K} h_K'^2 \|f + \Delta u_N\|^2_{L^2(K')} \quad \text{and} \quad \eta_2 = \sum_{e \in \mathcal{F}_K} h_e \|(\nabla u_N) \cdot n_e\|_{L^2(e)^2}$$

# 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

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

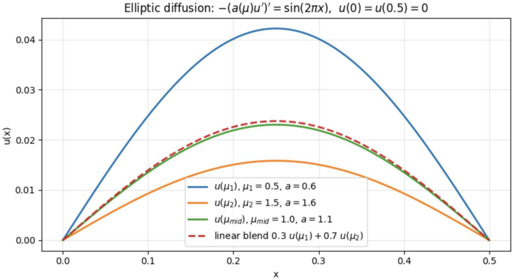
# Aposteriori

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

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

# TP Aposteriori

TP3: Kolmogorov very small




TP4: same PDE with a posteriori

# TP Kolmogorov

FEM with Kolmogorov $n$ width not small: Burgers equation:
Viscous Burgers (1D) with periodic BC using scikit-fem

$$\begin{aligned} u_t + \nu u u_x - \varepsilon u_x x = 0 \; in \; (0, T] \times [-1, 1] \\ u(0, x) = u_0(x) = \lambda + sin(x) \\ u \; periodic \; at \; x = -1 \; and \; x = 1 \end{aligned}$$

Time stepping: IMEX (explicit convection, implicit diffusion)
$(M + dt\varepsilon \; K)u^{n+1} = Mu^n - dtF(u^n)$ where $F_i(u^n) = \int \nu v_i(u^n)(u^n)_x dx$