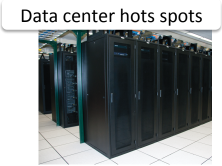
A non intrusive reduced basis method for heat transfer problem

Rachida CHAKIR

in collaboration with Yvon Maday, Pascal Joly and Philippe Parnaudeau

ENERGY SAVING

ENERGY EFFICIENTY



Needs

- 2D/3D Numerical Modeling
- Fast and reliable methods
- Control of quantity of interest
- Uncertainties quantifications

Challenges

- Multiphysics Modeling
- Non-Linearities and Coupling
- Complex geometries
- Optimization

Reduced basis methods

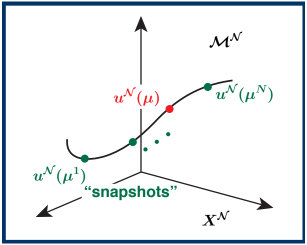
CONTEXT: optimization process or characterization in real-time of systems governed by a parameters dependent PDEs.

$\rightarrow$ Classicals discretization techniques such as finite element methods are generally too expensives

Given μ in $\mathcal{D} \subset \mathbb{R}^d$,

➡ Find $u^{\mathcal{N}}(\mu)$ in $X_{\mathcal{N}}$ such that

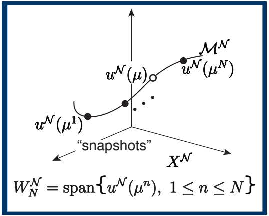
$$a(u^{\mathcal{N}}(\mu), v^{\mathcal{N}}; \mu) = f(v^{\mathcal{N}}; \mu)$$



$\rightarrow$ The reduced basis (R.B.) methods exploits the parametric structure of the governing PDE to construct rapidly convergent and computationally efficient approximations.

$\rightarrow$ Assume that $\mathcal{M}^{\mathcal{N}}(\mathcal{D}) = \{u^{\mathcal{N}}(\mu), \mu \in \mathcal{D}\}$ has a small (kolmogorov) dimension ...

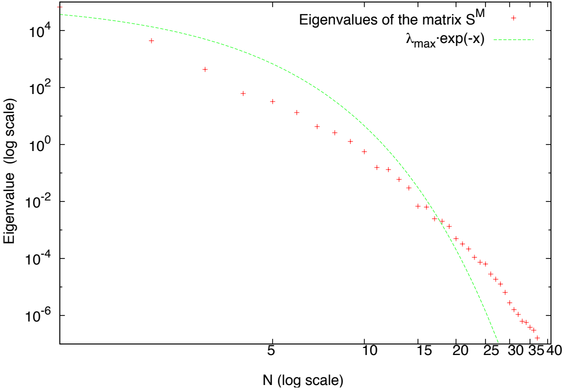
$\rightarrow$ we can select a set of parameters $(\mu^1, \cdots, \mu^N)$ in such way that $\mathcal{M}^{\mathcal{N}}(\mathcal{D})$ can be approximated by $W_N^{\mathcal{N}} = \text{span}\{u^{\mathcal{N}}(\mu^n), 1 \leq n \leq N\}$.



Evaluation of the dimension of $\mathcal{M}^{\mathcal{N}}(\mathcal{D})$?

Principal Analysis Component in appropriate norms :

$S^M_{k,\ell} = < u^{\mathcal{N}}(\mu_k), u^{\mathcal{N}}(\mu_\ell) >_X$, $1 \leq k, \ell \leq M$, M : number of snapshots



→ The R.B. method is based on the fact that for any $\varepsilon_N > 0$, there exist a set of parameters $(\mu_1, \cdots, \mu_N) \in \mathcal{D}^N$ such that:

$$\boxed{\forall \mu \in \mathcal{D}, \exists (\alpha_i(\mu)) \in \mathbb{R}^N, \quad \| u(\mu) - \sum_{i=1}^{N} \alpha_i(\mu)\, u(\mu^i) \|_{H^1(\Omega)} \le \varepsilon.}$$

→ The R.B. method is a Galerkin approach within the space $W_N^{\mathcal{N}}$.

THE REDUCED BASIS METHOD	vs	A CLASSICAL DISCRETIZATION METHOD
Find $u_N^h(\mu)$ in $W_N^{\mathcal{N}}$ s.t. : $a(u_N^h(\mu), v_N^h; \mu) = (f, v_N^h),$ $\forall v_N^h \in W_N^{\mathcal{N}}$ $\Rightarrow \mathcal{O}(N)$		Find $u^{\mathcal{N}}(\mu)$ in $X^{\mathcal{N}}$ s.t : $a(u^{\mathcal{N}}(\mu), v^h; \mu) = (f, v^h).$ $\forall v^h \in X^{\mathcal{N}}$ $\Rightarrow \mathcal{O}(\mathcal{N})$

→The reduced basis is promising if N is small ! $(N << \mathcal{N})$

Requirements of the reduced basis method:

- How to select the good sampling set $(\mu^1, \cdots, \mu^N)$?
  $\rightarrow$ Random
  $\rightarrow$ P.O.D
  $\rightarrow$ Greedy's algorithm

Algorithm 1 Example of a Greedy's algorithm

Given $\Xi_{train} = (\mu_1, \cdots, \mu_{n_{train}}) \in \mathcal{D}^{n_{train}}$, $n_{train} >> 1$
Choose randomly μ_1, $\rightarrow S_1 = \{\mu_1\}$ and $W_1^{\mathcal{N}} = \{u^{\mathcal{N}}(\mu_1)\}$
for $N = 2$ to N_{max} **do**
$\quad \mu_N = \arg\max_{\mu \in \Xi_{train}} \|u^{\mathcal{N}}(\mu) - u^h_{N-1}(\mu)\|_X$
$\quad S_N = S_{N-1} \cup \mu_N$ and $W_N^{\mathcal{N}} = W_{N-1}^{\mathcal{N}} + \mathrm{span}\{u^{\mathcal{N}}(\mu_N)\}$
end for

$\rightarrow$This version of the Greedy's algorithm is quite expensive !

Requirements of the reduced basis method:

- How to select the good sampling set $(\mu^1, \cdots, \mu^N)$?

  $\rightarrow$ Random

  $\rightarrow$ P.O.D

  $\rightarrow$ Greedy's algorithm

Algorithm 2 Example of a Greedy's algorithm

Given $\Xi_{train} = (\mu_1, \cdots, \mu_{n_{train}}) \in \mathcal{D}^{n_{train}}$, $n_{train} >> 1$
Choose randomly μ_1, $\rightarrow S_1 = \{\mu_1\}$ and $W_1^{\mathcal{N}} = \{u^{\mathcal{N}}(\mu_1)\}$
for $N = 2$ to N_{max} **do**

$$\mu_N = \arg \max_{\mu \in \Xi_{train}} \Delta_{N-1}(\mu)$$

$$S_N = S_{N-1} \cup \mu_N \text{ and } W_N^{\mathcal{N}} = W_{N-1}^{\mathcal{N}} + \mathrm{span}\{u^{\mathcal{N}}(\mu_N)\}$$

end for

$\Delta_N(\mu)$: sharp, inexpensive *a posteriori* error bound of $\|u^{\mathcal{N}}(\mu) - u_N^h(\mu)\|_X$
$\rightarrow$ Only the actual $u^{\mathcal{N}}(\mu_N)$ are computed by the Greedy's algorithm.

Requirements of the reduced basis method:

- How to select the good set of $(\mu^1, \cdots, \mu^N)$? (OFFLINE)
  $\rightarrow$ Random
  $\rightarrow$ "P.O.D"
  $\rightarrow$ Greedy algorithm
- How to actually computes the reduced solution $u_N^h(\mu)$ for a given μ?

$\rightarrow$ Get the classical solution $(u^{\mathcal{N}}(\mu^n))_{1\leq n\leq N}$ (for example using a FEM code), from which the orthogonal basis function $(\xi_1^{RB}, \cdots, \xi_N^{RB})$ of $W_N^{\mathcal{N}}$ will be computed.

For each new value of μ :

$\rightarrow$ build the matrix $[A^N(\mu)]_{k,\ell} = a(\xi_k^{RB}, \xi_\ell^{RB}; \mu)_{1\leq k,\ell\leq N}$ and the vector $[F^N(\mu)]_\ell = f(\xi_\ell^{RB}; \mu)_{1\leq \ell\leq N}$

$\rightarrow$ solve the system $A^N(\mu)\,\alpha^{N,h}(\mu) = F^N(\mu)$ and build output:

$$s(u_h^N(\mu)) = \sum_{\ell=1}^{N} \alpha_\ell^{N,h}(\mu)\, s(\xi_\ell^{BR})$$

$\rightarrow$ One of the keys of the R.B method is the decomposition of the computational work into an OFFLINE and an ONLINE stage

Requirements of the reduced basis method:

- How to select the good set of $(\mu^1, \cdots, \mu^N)$? (OFFLINE)
  $\rightarrow$ Random
  $\rightarrow$ "P.O.D"
  $\rightarrow$ Greedy algorithm
- How to actually computes the reduced solution $u_N^h(\mu)$ for a given μ?
  $\rightarrow$ Get the classical solution $(u^{\mathcal{N}}(\mu^n))_{1\leq n\leq N}$ (for example using a FEM code), from which the orthogonal basis function $(\xi_1^{RB}, \cdots, \xi_N^{RB})$ of $W_N^{\mathcal{N}}$ will be computed. (OFFLINE)

  For each new value of μ :
  $\rightarrow$ build the matrix $[A^N(\mu)]_{k,\ell} = a(\xi_k^{RB}, \xi_\ell^{RB}; \mu)_{1\leq k,\ell\leq N}$ and the vector $[F^N(\mu)]_\ell = f(\xi_\ell^{RB}; \mu)_{1\leq \ell\leq N}$ (OFFLINE + ONLINE)
  $\rightarrow$ solve the system $A^N(\mu)\,\alpha^{N,h}(\mu) = F^N(\mu)$ and build output:

  $$s(u_h^N(\mu)) = \sum_{\ell=1}^{N} \alpha_\ell^{N,h}(\mu)\, s(\xi_\ell^{BR}) \quad \text{(ONLINE)}$$

  $\rightarrow$ One of the keys of the R.B method is the decomposition of the computational work into an OFFLINE and an ONLINE stage

Requirements of the reduced basis method:

- How to select the good set of $(\mu^1, \cdots, \mu^N)$? (OFFLINE)
  $\rightarrow$ Random
  $\rightarrow$ "P.O.D"
  $\rightarrow$ Greedy algorithm
- How to actually computes the reduced solution $u_N^h(\mu)$ for a given μ?
  $\rightarrow$ Get the classical solution $(u^{\mathcal{N}}(\mu^n))_{1 \leq n \leq N}$ (for example using a FEM code), from which the orthogonal basis function $(\xi_1^{RB}, \cdots, \xi_N^{RB})$ of $W_N^{\mathcal{N}}$ will be computed. (OFFLINE)

  For each new value of μ :
  $\rightarrow$ build the matrix $[A^N(\mu)]_{k,\ell} = a(\xi_k^{RB}, \xi_\ell^{RB}; \mu)_{1 \leq k, \ell \leq N}$ and the vector $[F^N(\mu)]_\ell = f(\xi_\ell^{RB}; \mu)_{1 \leq \ell \leq N}$ (OFFLINE + ONLINE)
  $\rightarrow$ solve the system $A^N(\mu)\, \alpha^{N,h}(\mu) = F^N(\mu)$ and build output:

$$s(u_h^N(\mu)) = \sum_{\ell=1}^{N} \alpha_\ell^{N,h}(\mu)\, s(\xi_\ell^{BR}) \quad \text{(ONLINE)}$$

$\rightarrow$ One of the keys of the R.B method is the decomposition of the computational work into an OFFLINE and an ONLINE stage

Requirements of the reduced basis method:

- How to select the good set of $(\mu^1, \cdots, \mu^N)$? (OFFLINE)
  $\rightarrow$ Random
  $\rightarrow$ "P.O.D"
  $\rightarrow$ Greedy algorithm
- How to actually computes the reduced solution $u_N^h(\mu)$ for a given μ?
  $\rightarrow$ Get the classical solution $(u^{\mathcal{N}}(\mu^n))_{1\leq n\leq N}$ (for example using a FEM code), from which the orthogonal basis function $(\xi_1^{RB}, \cdots, \xi_N^{RB})$ of $W_N^{\mathcal{N}}$ will be computed. (OFFLINE)

  For each new value of μ :
  $\rightarrow$ build the matrix $[A^N(\mu)]_{k,\ell} = a(\xi_k^{RB}, \xi_\ell^{RB}; \mu)_{1\leq k,\ell\leq N}$ and the vector $[F^N(\mu)]_\ell = f(\xi_\ell^{RB}; \mu)_{1\leq \ell\leq N}$(OFFLINE + ONLINE)
  $\rightarrow$ solve the system $A^N(\mu)\,\alpha^{N,h}(\mu) = F^N(\mu)$ and build output:

  $$s(u_h^N(\mu)) = \sum_{\ell=1}^{N} \alpha_\ell^{N,h}(\mu)\, s(\xi_\ell^{BR}) \text{ (ONLINE)}$$

  $\rightarrow$ One of the keys of the R.B method is the decomposition of the computational work into an OFFLINE and an ONLINE stage

Requirements of the reduced basis method:

$\boxed{\text{How } A^N(\mu) \text{ is generated ?}}$

Direct affine's decomposition

$$a(\xi_i^{RB}, \xi_j^{RB}; \mu) = \sum_{k=1}^{\mathcal{K}} \theta_k(\mu)\, a_k(\xi_i^{RB}, \xi_j^{RB})$$

Empirical interpolation method

$$a(\xi_i^{RB}, \xi_j^{RB}; \mu) = \sum_{k=1}^{\mathcal{K}} \Phi_k(\mu)\, a(\xi_i^{RB}, \xi_j^{RB}; q_k)$$

OFFLINE: $a_k(\xi_i^{RB}, \xi_j^{RB})$ (or $a(\xi_i^{RB}, \xi_j^{RB}; q_k)$) are precomputed

ONLINE:
- $A^N(\mu)$ generation's requires only $\mathcal{K} \times N^2$ operations instead of $\mathcal{N}^2$.
- $A^N(\mu)$ inversion's is done in N^3 operations instead of $\mathcal{N}^3$.(direct inversion)

Generation of the output $s(u_h^N(\mu)) \Rightarrow s(\xi_i^{BR})$ also precomputed OFFLINE.

$\rightarrow$ All *expensive* computations are done in the OFFLINE stage

$\Rightarrow$ Then the ONLINE stage computations are in scale with N

Requirements of the reduced basis method:

$\boxed{\text{How } A^N(\mu) \text{ is generated ?}}$

Direct affine's decomposition

$$a(\xi_i^{RB}, \xi_j^{RB}; \mu) = \sum_{k=1}^{\mathcal{K}} \theta_k(\mu)\, a_k(\xi_i^{RB}, \xi_j^{RB})$$

Empirical interpolation method

$$a(\xi_i^{RB}, \xi_j^{RB}; \mu) = \sum_{k=1}^{\mathcal{K}} \Phi_k(\mu)\, a(\xi_i^{RB}, \xi_j^{RB}; q_k)$$

OFFLINE: $a_k(\xi_i^{RB}, \xi_j^{RB})$ (or $a(\xi_i^{RB}, \xi_j^{RB}; q_k)$) are precomputed

ONLINE:
- $A^N(\mu)$ generation's requires only $\mathcal{K} \times N^2$ operations instead of $\mathcal{N}^2$.
- $A^N(\mu)$ inversion's is done in N^3 operations instead of $\mathcal{N}^3$.(direct inversion)

Generation of the output $s(u_h^N(\mu)) \Rightarrow s(\xi_i^{BR})$ also precomputed OFFLINE.

→ All *expensive* computations are done in the OFFLINE stage

⇒ Then the ONLINE stage computations are in scale with N

Requirements of the reduced basis method:

$$\boxed{\text{How } A^N(\mu) \text{ is generated ?}}$$

Direct affine's decomposition

$$a(\xi_i^{RB},\xi_j^{RB};\mu) = \sum_{k=1}^{\mathcal{K}} \theta_k(\mu)\, a_k(\xi_i^{RB},\xi_j^{RB})$$

Empirical interpolation method

$$a(\xi_i^{RB},\xi_j^{RB};\mu) = \sum_{k=1}^{\mathcal{K}} \Phi_k(\mu)\, a(\xi_i^{RB},\xi_j^{RB};q_k)$$

OFFLINE: $a_k(\xi_i^{RB},\xi_j^{RB})$ (or $a(\xi_i^{RB},\xi_j^{RB};q_k)$) are precomputed

ONLINE:
- $A^N(\mu)$ generation's requires only $\mathcal{K}\times N^2$ operations instead of $\mathcal{N}^2$.
- $A^N(\mu)$ inversion's is done in N^3 operations instead of $\mathcal{N}^3$.(direct inversion)

Generation of the output $s(u_h^N(\mu)) \Rightarrow s(\xi_i^{BR})$ also precomputed OFFLINE.

$\rightarrow$ All *expensive* computations are done in the OFFLINE stage

$\Rightarrow$ Then the ONLINE stage computations are in scale with N

Requirements of the reduced basis method:

$\boxed{\text{How } A^N(\mu) \text{ is generated ?}}$

Direct affine's decomposition

$$a(\xi_i^{RB}, \xi_j^{RB}; \mu) = \sum_{k=1}^{\mathcal{K}} \theta_k(\mu)\, a_k(\xi_i^{RB}, \xi_j^{RB})$$

Empirical interpolation method

$$a(\xi_i^{RB}, \xi_j^{RB}; \mu) = \sum_{k=1}^{\mathcal{K}} \Phi_k(\mu)\, a(\xi_i^{RB}, \xi_j^{RB}; q_k)$$

OFFLINE: $a_k(\xi_i^{RB}, \xi_j^{RB})$ (or $a(\xi_i^{RB}, \xi_j^{RB}; q_k)$) are precomputed

ONLINE:
- $A^N(\mu)$ generation's requires only $\mathcal{K} \times N^2$ operations instead of $\mathcal{N}^2$.
- $A^N(\mu)$ inversion's is done in N^3 operations instead of $\mathcal{N}^3$.(direct inversion)

What happens when the FEM simulation code is used as black box ?

$\rightarrow$ It's not possible to use this code to perform all the OFFLINE computations required for an efficient performance of the R.B method

(since we want the online computation to be done with a N complexity and not with a complexity of the finite element method)

$\rightarrow$ An alternative : a non intrusive reduced basis method

A non intrusive reduced basis method : How ?

Let $\tilde{u}_h^N(\mu)$ be the L^2-projection of $u^{\mathcal{N}}(\mu)$ in $W_N^{\mathcal{N}}$ defined by

$$\boxed{\tilde{u}_h^N(\mu) = \sum_{i=1}^{N} \beta_i^{N,h}(\mu)\, \xi_i^{RB} \qquad \text{with} \qquad \beta_i^{N,h}(\mu) = \int_\Omega u^{\mathcal{N}}(\mu)\, \xi_i^{RB}}$$

$\longrightarrow$ The standard R.B. method aims at evaluating the coefficients $\alpha_i^{N,h}(\mu)$ those can appear as a substitute to the optimal coefficients $\beta_i^h(\mu)$.

Since, the computation of $u^{\mathcal{N}_H}(\mu)$, for $H >> h$ and $X_{\mathcal{N}_H} \subset X_{\mathcal{N}}$, is less expensive than the one of $u^{\mathcal{N}}(\mu)$.

$\longrightarrow$ Our alternative method [1,2] consists in proposing an another surrogate to $\beta_i^{N,h}(\mu)$ defined by

$$\boxed{\beta_i^{N,H}(\mu) = \int_\Omega u^{\mathcal{N}_H}(\mu)\, \xi_i^{RB}}$$

[1] R. Chakir Y. Maday, *A two-grid finite-element/reduced basis scheme for the approximation of the solution of parameter dependent P.D.E*, Actes de congrès du 9ème colloque national en calcul des structrures, Giens 2009.
[2] R. Chakir, Y. Maday, *Une méthode combinée d'éléments finis à deux grilles/bases réduites pour l'approximation des solutions d'une EDP paramétrique*, C. R. Acad. Sci. Paris, Ser. I 347 (2009) 435 - 440.

A non intrusive reduced basis method : How ?

Let $\tilde{u}_h^N(\mu)$ be the L^2-projection of $u^{\mathcal{N}}(\mu)$ in $W_N^{\mathcal{N}}$ defined by

$$\boxed{\tilde{u}_h^N(\mu) = \sum_{i=1}^{N} \beta_i^{N,h}(\mu)\,\xi_i^{RB} \qquad \text{with} \qquad \beta_i^{N,h}(\mu) = \int_\Omega u^{\mathcal{N}}(\mu)\,\xi_i^{RB}}$$

$\longrightarrow$ The standard R.B. method aims at evaluating the coefficients $\alpha_i^{N,h}(\mu)$ those can appear as a substitute to the optimal coefficients $\beta_i^h(\mu)$.

Since, the computation of $u^{\mathcal{N}_H}(\mu)$, for $H >> h$ and $X_{\mathcal{N}_H} \subset X_{\mathcal{N}}$, is less expensive than the one of $u^{\mathcal{N}}(\mu)$.

$\longrightarrow$ Our alternative method [1,2] consists in proposing an another surrogate to $\beta_i^{N,h}(\mu)$ defined by

$$\boxed{\beta_i^{N,H}(\mu) = \int_\Omega u^{\mathcal{N}_H}(\mu)\,\xi_i^{RB}}$$

[1] R. Chakir Y. Maday, *A two-grid finite-element/reduced basis scheme for the approximation of the solution of parameter dependent P.D.E*, Actes de congrès du 9ème colloque national en calcul des structures, Giens 2009.
[2] R. Chakir, Y. Maday, *Une méthode combinée d'éléments finis à deux grilles/bases réduites pour l'approximation des solutions d'une EDP. paramétrique*, C. R. Acad. Sci. Paris, Ser. I 347 (2009) 435 - 440.

We can build a reduced solution $u_{H,h}^{N}(\mu)$ and the output $s(u_{H,h}^{N}(\mu))$:

$$\boxed{u_{H,h}^{N}(\mu) = \sum_{i=1}^{N} \beta_i^{N,H}(\mu)\, \xi_i^{RB}} \quad \text{and} \quad \boxed{s(u_{H,h}^{N}(\mu)) = \sum_{i=1}^{N} \beta_i^{N,H}(\mu)\, s(\xi_i^{RB})}$$

$\rightarrow$ This method is based on the fact that the error measured in the L^2-norm converge faster than the one measured H^1-norm.

Why this can still be a good approximation?

$\longrightarrow$ The basis functions ξ_i^{RB} have to be orthonormal in H^1 and L^2 norm.

$X^{\mathcal{N}}$ and $X_H^{\mathcal{N}}$: $\mathbb{P}^k$- F.E discretization space $\rightarrow \|u(\mu) - u^{\mathcal{N}}(\mu)\|_X \leq c(\mu)\,\mathrm{h}^{\mathrm{k}}$

$\longrightarrow$ Using the orthogonality of ξ_i^{BR}, we easily can prove that :

$$\boxed{\|u(\mu) - u_{H,h}^{N}(\mu)\|_X \leq \varepsilon + C(\mu)\,(\mathrm{h}^{\mathrm{k}} + \mathrm{H}^{2\mathrm{k}})}$$

which is asymptotically similar to $\|u(\mu) - u_h^{N}(\mu)\|_X \leq \varepsilon + C(\mu)\,\mathrm{h}^{\mathrm{k}}$ when we choose $h \sim H^2$.

We can build a reduced solution $u^N_{H,h}(\mu)$ and the output $s(u^N_{H,h}(\mu))$:

$$\boxed{u^N_{H,h}(\mu) = \sum_{i=1}^{N} \beta_i^{N,H}(\mu)\,\xi_i^{RB}} \quad \text{and} \quad \boxed{s(u^N_{H,h}(\mu)) = \sum_{i=1}^{N} \beta_i^{N,H}(\mu)\,s(\xi_i^{RB})}$$

$\rightarrow$ This method is based on the fact that the error measured in the L^2-norm converge faster than the one measured H^1-norm.

Why this can still be a good approximation?

$\longrightarrow$ The basis functions ξ_i^{RB} have to be orthonormal in H^1 and L^2 norm.

$X^{\mathcal{N}}$ and $X_H^{\mathcal{N}}$: $\mathbb{P}^k$- F.E discretization space $\rightarrow \|u(\mu) - u^{\mathcal{N}}(\mu)\|_X \leq c(\mu)\,\mathrm{h}^{\mathrm{k}}$

$\longrightarrow$ Using the orthogonality of ξ_i^{BR}, we easily can prove that :

$$\boxed{\|u(\mu) - u^N_{H,h}(\mu)\|_X \leq \varepsilon + C(\mu)\,(\mathrm{h}^{\mathrm{k}} + \mathrm{H}^{2\mathrm{k}})}$$

which is asymptotically similar to $\|u(\mu) - u^N_h(\mu)\|_X \leq \varepsilon + C(\mu)\,\mathrm{h}^{\mathrm{k}}$ when we choose $h \sim H^2$.

Post-process to improve the computation of the $\beta_i^{N,H}(\mu)$

We computes the matrix $T^N \in \mathbb{R}^{N\times N}$ solution of the following system:

$$T^N \times \begin{pmatrix} \beta_1^{N,H}(\mu_1) & \cdots & \beta_1^{N,H}(\mu_N) \\ \cdots & \cdots & \cdots \\ \cdots & \cdots & \cdots \\ \beta_N^{N,H}(\mu_1) & \cdots & \beta_N^{N,H}(\mu_N) \end{pmatrix} = \begin{pmatrix} \beta_1^{N,h}(\mu_1) & \cdots & \beta_1^{N,h}(\mu_N) \\ \cdots & \cdots & \cdots \\ \cdots & \cdots & \cdots \\ \beta_N^{N,h}(\mu_1) & \cdots & \beta_N^{N,h}(\mu_N) \end{pmatrix}$$

$\Rightarrow$ We replace $u_{H,h}^N(\mu)$ and $s(u_{H,h}^N(\mu))$:

$$\boxed{\tilde{u}_{H,h}^N(\mu) = \sum_{i=1}^{N} T_{ij}^N \, \beta_i^{N,H}(\mu) \, \xi_i^{RB}} \quad \text{and} \quad \boxed{s(\tilde{u}_{H,h}^N(\mu)) = \sum_{i,j=1}^{N} T_{ij}^N \, \beta_i^{N,H}(\mu) \, s(\xi_i^{RB})}$$

What do we need ?

F.E. code used as *black box*

Compute
snapshots $u_h(\mu_i)$
coarse solution $u_H(\mu)$

Return
fine mesh $\mathcal{T}_h$
coarse mesh $\mathcal{T}_H$

F.E. library

(Freefem++)

To compute
L^2 and H^1
scalar product

Interpolate
from $\mathcal{T}_H$
to $\mathcal{T}_h$

Main characteristic of Freefem++[1]

→ Wide range of finite elements : continuous P1, P2 elements, discontinuous P0, P1, RT0, RT1, BDM1 elements, vectorial elements, ...

→ Automatic interpolation of data from a mesh to an other one (with matrix construction if need), so a finite element function is view as a function of (x; y; z) or as an array.

→ Link with other soft : paraview, gmsh, vtk, medit, gnuplot, ...

→ Dynamic linking to add plugin.

→ Full MPI interface

[1] http://www.freefem.org/ff++/

Implementation

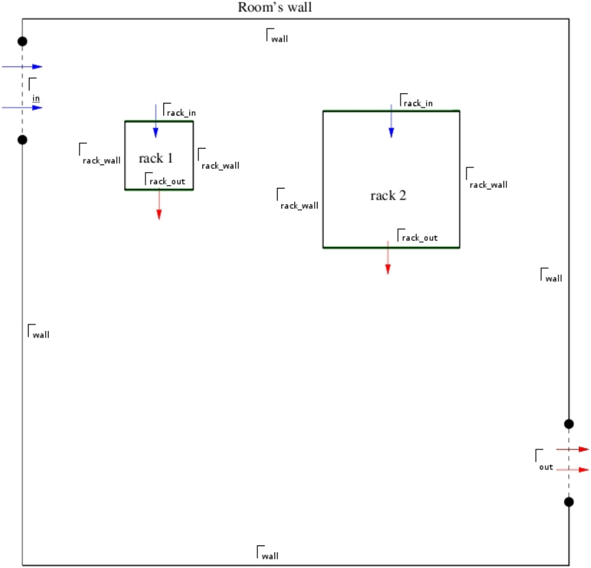
Offline stage

1. Construction of a reduced approximation's space.
   - computation of a sample of solutions (black box software)
   - selection of N solutions to build the reduced basis (F.E. Library).
2. Orthonormalisation in L^2 and H^1-norm of the reduced basis functions (F.E. Library).
3. Preparation for the post-processing.
   - computation of the N coarse solutions $u^{\mathcal{N}_H}(\mu_i)$ (black box software)
   - construction of matrix T^N (F.E. Library).

Online stage

1. Computation of the coarse solution $u^{\mathcal{N}_H}(\mu)$.(black box software)
2. Compute the coefficient $\beta_i^{N,H}(\mu)$. (F.E. Library)
3. Apply the post-processing on the $\beta_i^{N,H}(\mu)$. (F.E. Library)
4. Build the output $s(u_N^{H,h}(\mu))$. (F.E. Library)

Model : Incompressible steady Navier Stokes + Heat equation (Boussinesq's approximation)



Domain's definition of the simplified data center room

Cooling air velocity
$V_{in} \in [0.5; 2]$,

Cooling air temperature
$\theta_{in} \in [288; 292]$,

Rack's air velocity
$V_{rack} \in [0.1; 0.4]$,

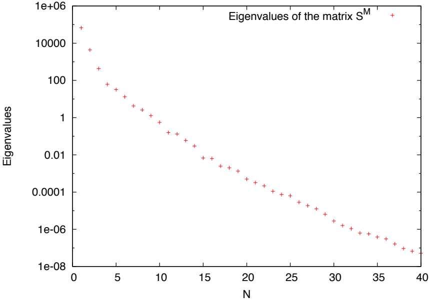
Rack's air temperature
$\theta_{rack} \in [295; 315]$.

Sampling to extract the reduced basis

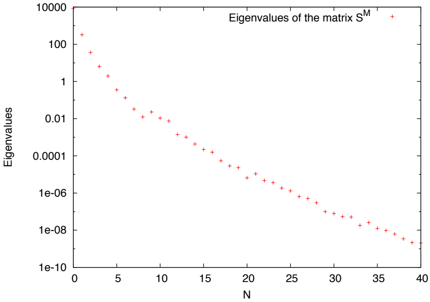
θ_{in}	V_{in}	V_{rack}	θ_{rack}
288	0.5	0.1	295
292	1	0.2	300
	2	0.3	305
		0.4	310
			315

Computations of 120 snapshots using a $\mathbb{P}_2 - \mathbb{P}_1$ F.E steady Navier-Stokes solver within Freefem++ on a reference mesh.

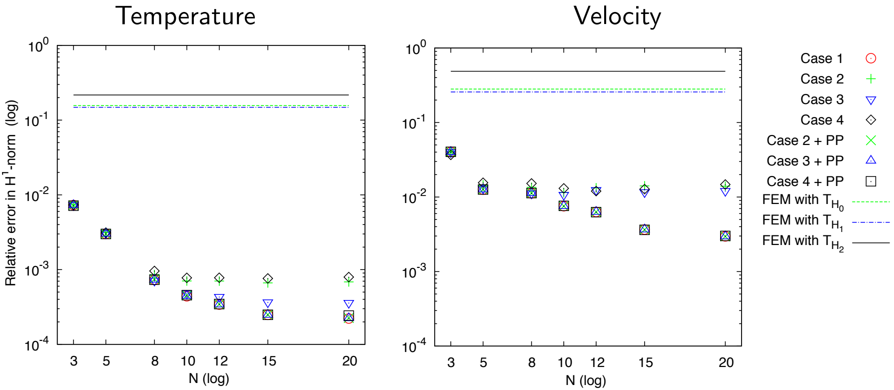
Figure: Values of the N largest eigenvalues of the matrix $S^{\mathcal{M}}$

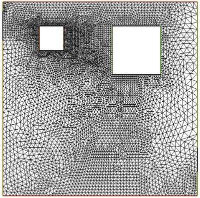


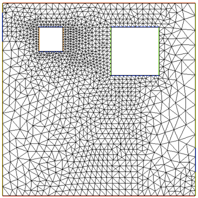
Temperature

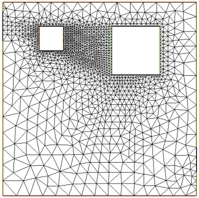


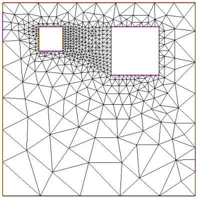
Velocity




reference mesh
Ndof = 20213
Case 1

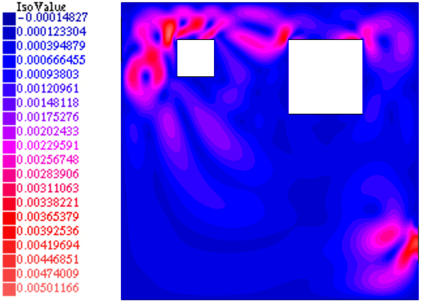

coarse embedded mesh
Ndof = 5148
Case 2


coarse non embedded mesh
Ndof = 5088
Case 3

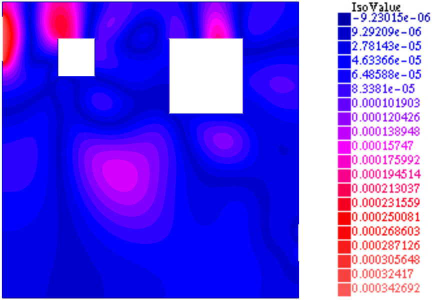

very coarse non embedded mesh
Ndof = 1871
Case 4

Relative error plot between the reference F.E. and the NIRB solutions

(case 4 + p.p. with $N = 15$)



Velocity magnetude



Tempertaure

Mean value of the online's stage with post-processing executions's time - $N = 15$

Reference FEM	NIRB - case 2	NIRB - case 3	NIRB - case 4	
200 sec	52 sec	52 sec	17 sec	Temperature
	53 sec	53 sec	18 sec	Velocity
	54 sec	54 sec	19 sec	Both

Conclusion

We note that the post-processing improved even more the approximation since it allows to recover the truth error even starting from the computations of the coarsest NIRB solution.

Perspectives

Apply to more complex application:

$\longrightarrow$ time dependent problem

$\longrightarrow$ take geometry as a parameter