

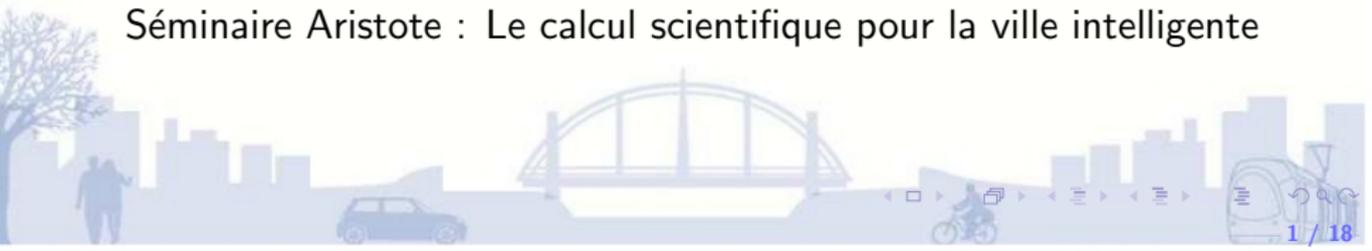
# *Modélisation pour la qualité d'eau dans les réseaux d'eau potable*

Rachida Chakir

Laboratoire Instrumentation, Simulation et Informatique Scientifique



Séminaire Aristote : Le calcul scientifique pour la ville intelligente





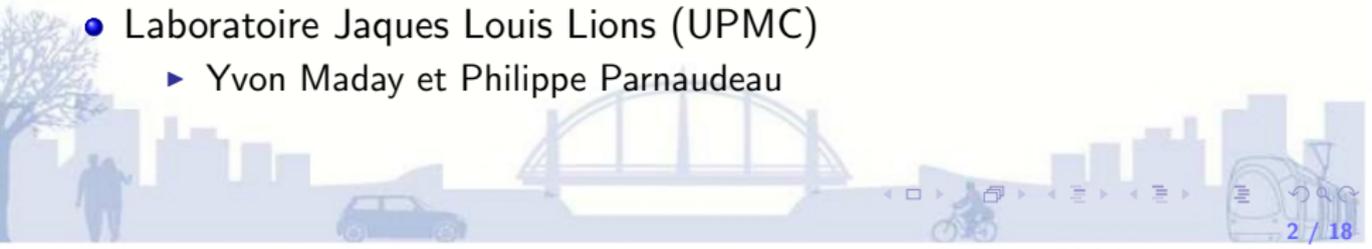
la ville intelligente et communicante



Qualité de l'eau

## Travaux en collaboration avec

- Laboratoire Instrumentation, Simulation et Informatique Scientifique (IFSTTAR - Département Composants et Systèmes)
  - ▶ Frédéric Bourquin, Patrice Chatellier, Erick Merliot et Julien Waeytens
- Laboratoire Jaques Louis Lions (UPMC)
  - ▶ Yvon Maday et Philippe Parnaudeau





la ville intelligente et communicante



Qualité de l'eau

## But

- Reconstruction des champs de vitesse d'un écoulement et de concentration de polluants, grâce à des données issues de capteur.
  - ▶ Calibrage des modèles mathématiques
  - ▶ Prédiction de l'évolution de la qualité de l'eau et détection d'anomalies

## Besoins

- Simulation numérique rapide et fiable

## Méthodes inverses et calibration de modèle pour la qualité de l'eau

**But** : retrouver le ou les paramètres inconnus d'un modèle

**Comment** : On définit d'une fonctionnelle mesurant l'écart entre mesure des capteurs et le modèle, pour un lot de paramètres  $\sigma$  donné.

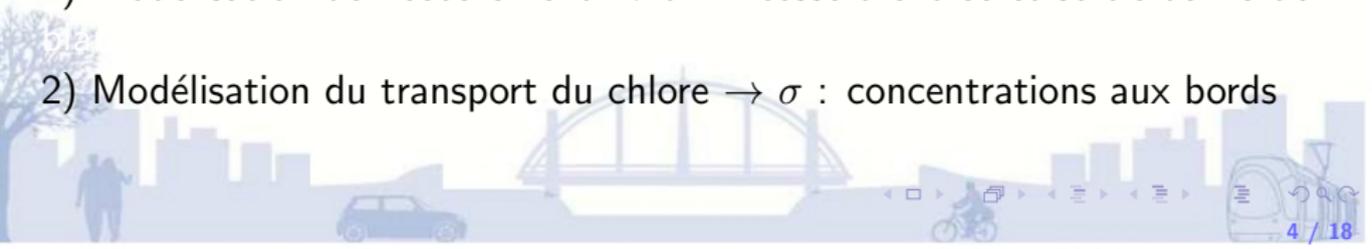
$$J(\sigma) = \frac{1}{2} \|W_{sim}(x; \sigma) - W_{mes}\|^2$$

Puis on cherche les paramètres  $\sigma$  optimaux minimisant la fonctionnelle  $J$ .

→ Résolution du problème de minimisation sous contraintes par un algorithme de descente avec l'état adjoint

1) Modélisation de l'écoulement →  $\sigma$  : vitesse d'entrée et sortie du fluide

2) Modélisation du transport du chlore →  $\sigma$  : concentrations aux bords



## Les équations de Navier-Stokes pour un écoulement incompressible

$$\rho \left( \overbrace{\frac{\partial \mathbf{v}}{\partial t} + (\mathbf{v} \cdot \nabla) \mathbf{v}}^{\text{forces d'accélération}} \right) + \underbrace{\nabla p - \mu \Delta \mathbf{v}}_{\text{forces internes}} = \mathbf{0}, \quad \text{div } \mathbf{v} = 0, \quad + \text{Cond. Lim}$$

## Modèles simplifiés

- Les équations de Stokes stationnaire

$$\rho \left( \cancel{\frac{\partial \mathbf{v}}{\partial t} + (\mathbf{v} \cdot \nabla) \mathbf{v}} \right) + \nabla p - \mu \Delta \mathbf{v} = \mathbf{0}, \quad \text{div } \mathbf{v} = 0, \quad + \text{Cond. Lim}$$

- Les équations de Stokes instationnaire

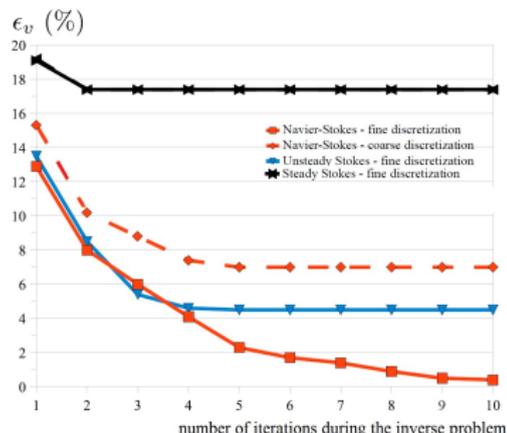
$$\rho \left( \frac{\partial \mathbf{v}}{\partial t} + \cancel{(\mathbf{v} \cdot \nabla) \mathbf{v}} \right) + \nabla p - \mu \Delta \mathbf{v} = \mathbf{0}, \quad \text{div } \mathbf{v} = 0, \quad + \text{Cond. Lim}$$

## Qualité de l'écoulement reconstruit sur une jonction en Té [1,2]

$\epsilon_V$  : Erreur relative entre l'écoulement 2D de référence et l'écoulement reconstruit par le modèle inverse



- Navier Stokes  $\rightarrow \epsilon_V < 1\%$   
Maillage fin  $\rightarrow 100$  min/iteration  
Maillage grossier  $\rightarrow 2$  min/iteration
- Stokes instationnaire  $\rightarrow \epsilon_V \approx 5\%$   
Maillage fin  $\rightarrow 2$  min/iteration
- Stokes stationnaire  $\rightarrow \epsilon_V \approx 17\%$   
Maillage fin  $< 1$  sec/iteration



Paramètres à identifier  $\sigma$  : vitesse d'entrée et sortie du fluide

[1] J. Waeytens, P. Chatellier, F. Bourquin, *Inverse computational fluid dynamics: influence of discretisation and model errors on flow in water network including junctions*, Fluids Engineering, (accepted with minor revisions).

[2] J. Waeytens, P. Chatellier, F. Bourquin, *Influence of discretization and model errors on inverse computational fluid: application to a water pipe junction*, ECCOMAS, Vienna (Austria), 2012.

## Besoins

- Un outil peu coûteux, rapide et fiable

### Modèles 1D

- ⊕ Calcul court et peu coûteux
- ⊖ Le mélange du chlore dans les jonctions est supposé complet et instantané

### Modèles 2D/3D

- ⊕ Meilleure description de l'écoulement et du chlore dans les jonctions
- ⊖ Calcul long et coûteux



### Réduction de modèles

- ▶ Méthodes des bases réduites



## Calibration ou caractérisation en temps réel de systèmes décrit par des EDPs dépendant de paramètres

→ Les méthodes de discrétisation classique sont souvent trop coûteuses.

Pour  $\sigma$  dans  $\mathcal{D} \subset \mathbb{R}^d$  donné

→ Trouver  $u_h(\sigma)$  dans  $\mathbb{R}^{\mathcal{N}_h}$  tq

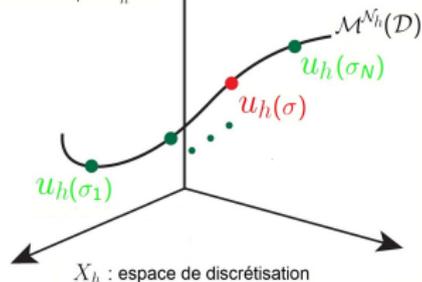
$$A^{\mathcal{N}_h}(\sigma) u_h(\sigma) = F^{\mathcal{N}_h}(\sigma)$$

$\sigma$  : jeu de paramètres (conditions limites, paramètres physiques, ...)

$$\mathcal{M}^{\mathcal{N}_h}(\mathcal{D}) \equiv \{u_h(\sigma) \mid \sigma \in \mathcal{D}\}$$

$\mathcal{D}$  : espace des paramètres

$\mathcal{N}_h$  : dimension de l'espace  $X_h$



- On peut choisir un ensemble de paramètres  $(\sigma_1, \dots, \sigma_N)$  tel que l'ensemble des solutions  $\mathcal{M}^{\mathcal{N}_h}(\mathcal{D}) \equiv \{u_h(\sigma), \sigma \in \mathcal{D}\}$  puisse être approché par l'espace  $X_h^{\mathcal{N}} \equiv \{u_h(\sigma_n), 1 \leq n \leq N\}$  appelé espace des « bases réduites ».

Pour  $\sigma \in \mathcal{D}$  donné

Méthodes classiques (EF, VF, ...)

Trouver  $u_h(\sigma)$  dans  $\mathbb{R}^{\mathcal{N}_h}$  tq

$$A^{\mathcal{N}_h}(\sigma) u_h(\sigma) = F^{\mathcal{N}_h}(\sigma)$$

→ Résolution en  $\mathcal{N}_h^3$  opérations

VS

Méthodes des bases réduites

Trouver  $u_h^N(\sigma)$  dans  $\mathbb{R}^N$  tq

$$A^N(\sigma) u_h^N(\sigma) = F^N(\sigma)$$

→ Résolution en  $N^3$  opérations

→ La méthode des bases réduites est prometteuse lorsque  $N$  est petit !  
( $N \ll \mathcal{N}_h$ )



## Prérequis pour la méthode des bases réduites:

- Choisir l'ensemble  $(\sigma^1, \dots, \sigma^N)$ 
  - Au hasard
  - Decomposition orthogonale propre (POD)
  - Méthode « glouton » (Greedy algorithm)
- A partir des  $N$  solutions  $(u_h(\sigma^n))_{1 \leq n \leq N}$  (obtenu via un code EF ou VF par exemple), calculer les fonctions de bases orthogonales  $(\xi_1^N, \dots, \xi_N^N)$  de  $X_h^N$ .
- Pour chaque nouvelle valeur de  $\sigma$ , mettre à jour  $A_N(\sigma)$  et  $F_N(\sigma)$



## Prérequis pour la méthode des bases réduites:

- Choisir l'ensemble  $(\sigma^1, \dots, \sigma^N)$ 
  - Au hasard
  - Decomposition orthogonale propre (POD)
  - Méthode « glouton » (Greedy algorithm)
- A partir des **N solutions**  $(u_h(\sigma^n))_{1 \leq n \leq N}$  (obtenu via un code EF ou VF par exemple), calculer les fonctions de **bases orthogonales**  $(\xi_1^N, \dots, \xi_N^N)$  de  $X_h^N$ .
- Pour chaque nouvelle valeur de  $\sigma$  : mettre à jour  $A_N(\sigma)$  et  $F_N(\sigma)$

$$A_N(\sigma) = \begin{pmatrix} \xi_1^N \\ \vdots \\ \xi_N^N \end{pmatrix} A_N(\sigma) (\xi_1^N, \dots, \xi_N^N)$$

→ La mise à jour de  $A_N(\sigma)$  et  $F_N(\sigma)$  serait trop coûteuse!  
( $N_h^2$  opérations)



## Prérequis pour la méthode des bases réduites:

- Choisir l'ensemble  $(\sigma^1, \dots, \sigma^N)$ 
  - Au hasard
  - Decomposition orthogonale propre (POD)
  - Méthode « glouton » (Greedy algorithm)
- A partir des **N solutions**  $(u_h(\sigma^n))_{1 \leq n \leq N}$  (obtenu via un code EF ou VF par exemple), calculer les fonctions de **bases orthogonales**  $(\xi_1^{\mathcal{N}_h}, \dots, \xi_N^{\mathcal{N}_h})$  de  $X_h^{\mathcal{N}}$ .
- Pour chaque **nouvelle valeur de  $\sigma$**  : mettre à jour  $A_N(\sigma)$  et  $F_N(\sigma)$

~~$$A_N(\sigma) = \begin{pmatrix} \xi_1^{\mathcal{N}_h} \\ \vdots \\ \xi_N^{\mathcal{N}_h} \end{pmatrix} A_{\mathcal{N}_h}(\sigma) (\xi_1^{\mathcal{N}_h}, \dots, \xi_N^{\mathcal{N}_h})$$~~

→ La mise à jour de  $A_N(\sigma)$  et  $F_N(\sigma)$  serait trop coûteuse!  
 $(\mathcal{N}_h^2$  opérations)



## Prérequis pour la méthode des bases réduites:

- Choisir l'ensemble  $(\sigma^1, \dots, \sigma^N)$ 
  - Au hasard
  - Decomposition orthogonale propre (POD)
  - Méthode « glouton » (Greedy algorithm)
- A partir des **N solutions**  $(u_h(\sigma^n))_{1 \leq n \leq N}$  (obtenu via un code EF ou VF par exemple), calculer les fonctions de **bases orthogonales**  $(\xi_1^{\mathcal{N}_h}, \dots, \xi_N^{\mathcal{N}_h})$  de  $X_h^{\mathcal{N}}$ .
- Pour chaque **nouvelle valeur de  $\sigma$**  : mettre à jour  $A_N(\sigma)$  et  $F_N(\sigma)$

~~$$A_N(\sigma) = \begin{pmatrix} \xi_1^{\mathcal{N}_h} \\ \vdots \\ \xi_N^{\mathcal{N}_h} \end{pmatrix} A_{\mathcal{N}_h}(\sigma) \begin{pmatrix} \xi_1^{\mathcal{N}_h}, \dots, \xi_N^{\mathcal{N}_h} \end{pmatrix}$$~~

→ La mise à jour de  $A_N(\sigma)$  et  $F_N(\sigma)$  serait trop coûteuse!  
 $(\mathcal{N}_h^2$  opérations)



## Prérequis pour la méthode des bases réduites:

- Choisir l'ensemble  $(\sigma^1, \dots, \sigma^N)$ 
  - Au hasard
  - Decomposition orthogonale propre (POD)
  - Méthode « glouton » (Greedy algorithm)
- A partir des **N solutions**  $(u_h(\sigma^n))_{1 \leq n \leq N}$  (obtenus via un code EF ou VF par exemple), calculer les fonctions de **bases orthogonales**  $(\xi_1^{\mathcal{N}_h}, \dots, \xi_N^{\mathcal{N}_h})$  de  $X_h^{\mathcal{N}}$ .
- Pour chaque **nouvelle valeur** de  $\sigma$  : mettre à jour  $A_N(\sigma)$  et  $F_N(\sigma)$

$$A_N(\sigma) = \sum_{k=1}^M \theta_k(\sigma) \underbrace{\left[ \begin{pmatrix} \xi_1^{\mathcal{N}_h} \\ \vdots \\ \xi_N^{\mathcal{N}_h} \end{pmatrix} A_{\mathcal{N}_h}^k \left( \xi_1^{\mathcal{N}_h}, \dots, \xi_N^{\mathcal{N}_h} \right) \right]}_{= A_N^k(\sigma)}$$

- Une des clés des méthodes des bases réduites est le découpage de la mise en œuvre en une étape **OFFLINE** de précalcul, puis pour chaque nouvelle valeur de  $\sigma$  l'étape **ONLINE**

## Prérequis pour la méthode des bases réduites:

- Choisir l'ensemble  $(\sigma^1, \dots, \sigma^N)$ 
  - Au hasard
  - Decomposition orthogonale propre (POD)
  - Méthode « glouton » (Greedy algorithm)
- A partir des **N solutions**  $(u_h(\sigma^n))_{1 \leq n \leq N}$  (obtenus via un code EF ou VF par exemple), calculer les fonctions de **bases orthogonales**  $(\xi_1^{\mathcal{N}_h}, \dots, \xi_N^{\mathcal{N}_h})$  de  $X_h^{\mathcal{N}}$ .
- Pour chaque **nouvelle valeur de  $\sigma$**  : mettre à jour  $A_N(\sigma)$  et  $F_N(\sigma)$

$$A_N(\sigma) = \sum_{k=1}^M \theta_k(\sigma) \underbrace{\left[ \begin{pmatrix} \xi_1^{\mathcal{N}_h} \\ \vdots \\ \xi_N^{\mathcal{N}_h} \end{pmatrix} A_{\mathcal{N}_h}^k \left( \xi_1^{\mathcal{N}_h}, \dots, \xi_N^{\mathcal{N}_h} \right) \right]}_{= A_N^k(\sigma)}$$

- Une des clés des méthodes des bases réduites est le découpage de la mise en œuvre en une étape **OFFLINE** de précalcul, puis pour chaque nouvelle valeur de  $\sigma$  l'étape **ONLINE**

## Prérequis pour la méthode des bases réduites:

- Choisir l'ensemble  $(\sigma^1, \dots, \sigma^N)$  (OFFLINE)
  - Au hasard
  - Decomposition orthogonale propre (POD)
  - Méthode « glouton » (Greedy algorithm)
- A partir des  $N$  solutions  $(u_h(\sigma^n))_{1 \leq n \leq N}$  (obtenu via un code EF ou VF par exemple), calculer les fonctions de bases orthogonales  $(\xi_1^N, \dots, \xi_N^N)$  de  $W_h^N$ . (OFFLINE)
- Pour chaque nouvelle valeur de  $\sigma$  : mettre à jour  $A_N(\sigma)$  et  $F_N(\sigma)$

$$A_N(\sigma) = \sum_{k=1}^M \overbrace{\theta_k(\sigma)}^{(\text{ONLINE})} \underbrace{\left[ \begin{pmatrix} \xi_1^N \\ \vdots \\ \xi_N^N \end{pmatrix} A_{\mathcal{N}_h}^k \left( \xi_1^N, \dots, \xi_N^N \right) \right]}_{= A_N^k (\text{OFFLINE})}$$

→  $kN^2$  opérations pour mettre à jour  $A_N(\sigma)$  durant la phase ONLINE

- Résolution du problème réduit  $A_N(\sigma) u_h^N(\sigma) = F_N(\sigma)$  (ONLINE)



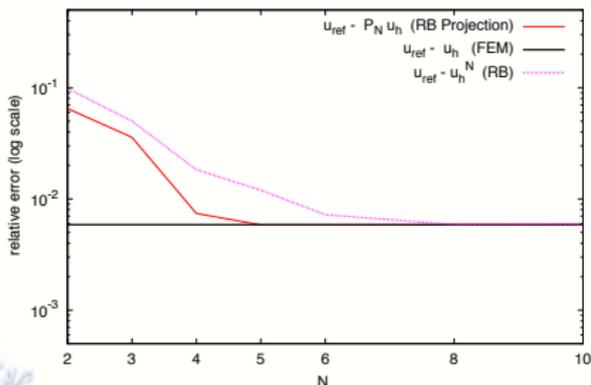
## Cas test jonction en T (Eq Navier Stokes 2D)

avec  $\sigma = v_{in} \in [0.01; 0.5] cm.s^{-1}$  : la vitesse d'entrée

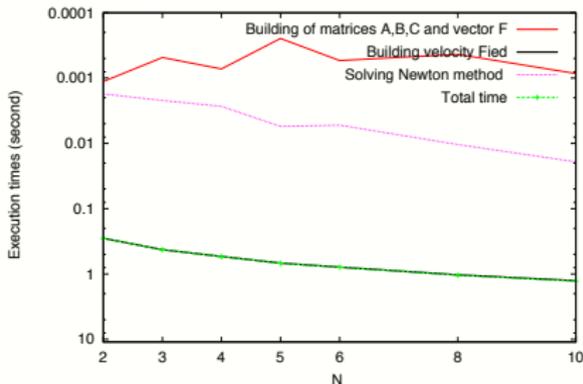
$Re \in [10, 500]$ ,  $Re = \frac{\rho D v_{in}}{\mu}$ , D: diamètre du tuyau,  $v_{in}$ : vitesse d'entrée,  $\rho$  : densité et  $\mu$ : viscosité dynamique.

$u_h \rightarrow \mathbb{P}_2$ -FE solution N dof = 34313,  $u_{ref} \rightarrow \mathbb{P}_2$ -FE solution N dof = 136145

Relative error between FEM solution and RB solution in norm L2



Times execution for the reduced problem



Méthode des éléments finis

640 sec

Méthode des bases réduites

4.2 sec (étape ONLINE)

Comment construire  $A_N(\sigma)$  lorsque la matrice  $A_{\mathcal{N}_h}(\sigma)$  est fortement non linéaire, ou lorsqu'on utilise un code verrouillé ou un logiciel commercial ?

$$A_N(\sigma) = \sum_{k=1}^M \overbrace{\theta_k(\sigma)}^{(\text{ONLINE})} \underbrace{\left[ \begin{pmatrix} \xi_1^{\mathcal{N}_h} \\ \vdots \\ \xi_N^{\mathcal{N}_h} \end{pmatrix} A_{\mathcal{N}_h}^k \left( \xi_1^{\mathcal{N}_h}, \dots, \xi_N^{\mathcal{N}_h} \right) \right]} = A_N^k (\text{OFFLINE})$$

→ Il est impossible d'accomplir tous les pré-calculs durant la phase OFFLINE pour une performance optimale de la méthode

(certaines opérations durant la phase ONLINE pourrait pas être de complexité  $\mathcal{N}_h$  au lieu de  $N$ )

→ Une alternative : une méthode des bases réduites non intrusive

## Une méthode des bases réduites non intrusive : Comment ?

$$P_N u_h(\sigma) = \sum_{i=1}^N \beta_i^{N,h}(\sigma) \xi_i^{\mathcal{N}_h}, \quad L^2\text{-projection de } u_h(\sigma) \text{ dans } X_h^N$$

→ La méthode des bases réduites consiste à approcher les coefficients  $\beta^{N,h}(\sigma)$  par  $u_h^N(\sigma)$  solution du système linéaire réduit.

Comme, le calcul de  $u_H(\sigma)$ , pour  $\mathcal{N}_H \ll \mathcal{N}_h$  et  $X_{\mathcal{N}_H} \subset X_{\mathcal{N}}$ , est bien coûteux que celui de  $u^{\mathcal{N}}(\sigma)$ .

→ Notre méthode non intrusive [1,2] consiste à approcher les coefficients  $\beta^{N,h}(\sigma)$  par

$$\gamma^{N,H}(\sigma) = T^N \beta^{N,H}(\sigma)$$

$$\text{avec } \beta_i^{N,H}(\sigma) = (u^{\mathcal{N}_H}(\sigma), \xi_i^{\mathcal{N}_h})_{L^2} \text{ et } T^N = \begin{pmatrix} \beta_1^{N,h}(\sigma_1) & \cdots & \beta_1^{N,h}(\sigma_N) \\ \vdots & \ddots & \vdots \\ \beta_N^{N,h}(\sigma_1) & \cdots & \beta_N^{N,h}(\sigma_N) \end{pmatrix} \times \begin{pmatrix} \beta_1^{N,H}(\sigma_1) & \cdots & \beta_1^{N,H}(\sigma_N) \\ \vdots & \ddots & \vdots \\ \beta_N^{N,H}(\sigma_1) & \cdots & \beta_N^{N,H}(\sigma_N) \end{pmatrix}^{-1}$$

[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.

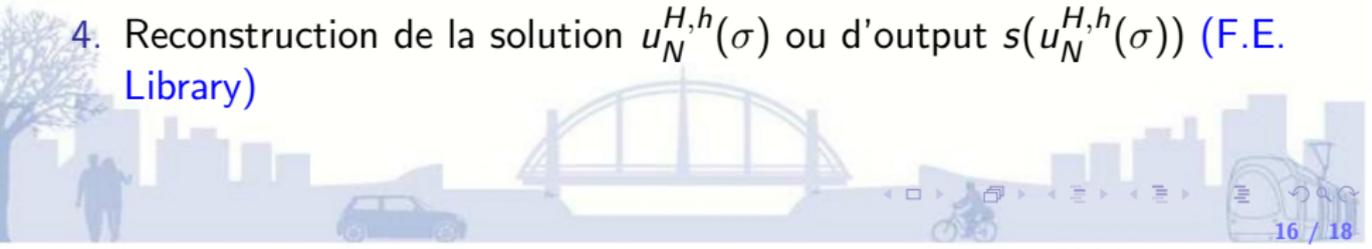
## MISE EN ŒUVRE

### OFFLINE stage

1. Construction de l'espace des bases réduites.
  - ▶ Calcul de  $N_{max}$  solutions particulières  $u_h(\sigma_i)$  (black box software)
  - ▶ Selection de  $N$  solutions  $u_h(\sigma_i)$  (F.E. Library).
2. Orthonormalisation des fonctions de bases (F.E. Library).
3. Pre-calcul pour la rectification.
  - ▶ calcul des  $N$  solutions grossières  $u_H(\sigma_i)$  (black box software)
  - ▶ construction de la matrice  $T^N$  (F.E. Library).

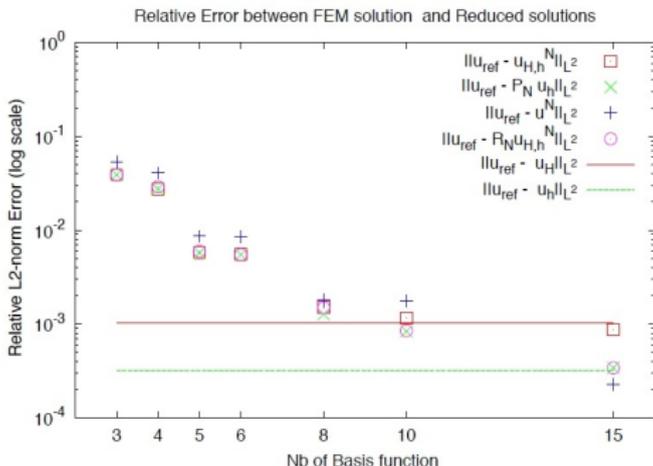
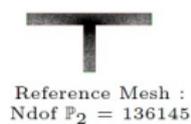
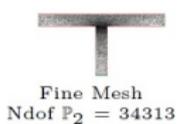
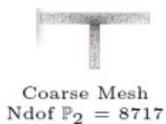
### ONLINE stage

1. Calcul de la solution grossière  $u^{N_H}(\sigma)$ . (black box software)
2. Calcul des coefficients  $\gamma_i^{N,H}(\sigma)$ . (F.E. Library)
4. Reconstruction de la solution  $u_N^{H,h}(\sigma)$  ou d'output  $s(u_N^{H,h}(\sigma))$  (F.E. Library)



# Cas test jonction en T (Eq N.S 2D) avec $\sigma = v_{in} \in [0.01; 0.5] cm.s^{-1}$

$Re \in [10, 500]$ ,  $Re = \frac{\rho D v_{in}}{\mu}$ , D: diamètre du tuyau,  $v_{in}$ : vitesse d'entrée,  $\rho$ : densité et  $\mu$ : viscosité dynamique.



Méthode des éléments finis

640 sec

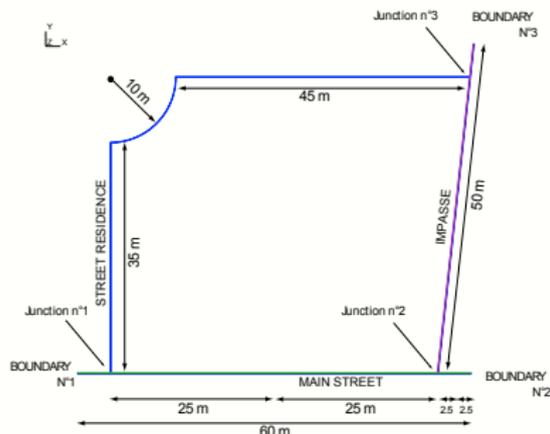
Méthode des bases réduites

60 sec (étape ONLINE)

## Reconstruction du champs de concentration de chlore [1]

L'écoulement connu  $v$  est turbulent, Vidéo reconstruction chlore

Objectif PB Inverse: On cherche à déterminer la concentration  $\sigma$  aux bords du domaine B2 (à l'entrée) minimisant l'écart entre simulation et mesure.



$$\frac{\partial C}{\partial t} + v \cdot \nabla C - \nu \Delta C + kC = 0$$

$$C(t = 0) = C_0$$

+ cond. Lim.

## Perspectives

- Application de réduction de modèle pour la reconstruction d'un écoulement à l'échelle du quartier

→ Code\_Saturne pour modéliser les écoulements plus complexe

[1] J. Waeytens, P. Chatellier, F. Bourquin, *Sensitivity of inverse advection-diffusion-reaction to sensor and control: a low computational cost tool*, Computers and Mathematics with Applications, Vol. 66, Issue 6, 2013.