



ÉCOLE POLYTECHNIQUE
FÉDÉRALE DE LAUSANNE

Introduction à la méthode des bases réduites

Gwenol Grandperrin

Assistant : Gianluigi Rozza

Professeur : Alfio Quarteroni

Semestre d'automne 2008-2009

Table des matières

1	Introduction	1
2	Résultats d'analyse fonctionnelle	1
3	Problématique	2
3.1	Définition du problème	2
3.2	Restriction sur $a(\cdot, \cdot; \mu)$	2
3.3	Approximation par éléments finis	3
4	Représentations géométriques affines	3
5	Considération sur les bases réduites	4
5.1	Base réduite de Taylor	4
5.2	Base réduite de Lagrange	5
5.3	Généralisation	5
6	Méthode des bases réduites	5
6.1	Espace des bases réduites	5
6.2	Algorithme glouton (« greedy ») pour la construction des espaces des bases réduites de Lagrange	6
6.3	Projection de Galerkin	6
6.4	Procédure Online-Offline	7
7	Borne inférieure pour la constante de coercivité	8
7.1	Méthode des Contraintes successives	8
7.2	Discussion	10
8	Borne supérieure pour la constante de coercivité	10
8.3	Discussion	11
8.4	Algorithme gloutin de sélection des C_J	11
8.5	Procédure de calcul « offline-online »	12
9	Estimation de l'erreur a posteriori	13
9.1	Borne de l'erreur	14
9.2	Approche pratique pour calculer l'erreur	16
9.3	Procédure de calcul « offline-online »	17
10	Mise en pratique de la méthode des bases réduites	17
10.1	Le logiciel <code>rbMIT</code>	18
10.2	Outils de visualisation de résultats	18
10.3	Modélisation d'un domaine sous <code>rbMIT</code>	18
10.4	Un premier modèle	19
10.5	Un modèle plus élaboré	24
10.6	Écoulement autour d'un fuselage	28
10.7	Modèles et efforts de calcul	33
11	Conclusion	34
A	Fichiers sources pour les modèles proposés	35

1 Introduction

Que ce soit pour des raisons de coût ou de temps, il n'est plus envisageable à l'heure actuelle de se passer de simulations numériques. Les ingénieurs d'aéronautique, d'automobile ou autres l'ont bien compris. Pour tester leurs nouveaux prototypes ils ont souvent recours à des calculs numériques utilisant des méthodes de différences finis ou des éléments finis.

Ces méthodes numériques ont fait leurs preuves depuis des années déjà. Cependant, dans le domaine des calculs ou des simulations en temps réel, elles ne parviennent en général pas à fournir un résultat dans un temps acceptable. La méthode des bases réduites fournit une réponse à ce problème. Elle permet de considérer un problème dont la solution dépendant de paramètres. Ainsi partant d'une valeur pour ces paramètres, elle permet d'obtenir la solution correspondante après un temps de calcul relativement court. Bien entendu pour parvenir à un tel résultat, il est crucial de calculer et stocker certaines quantités qui seront utilisées lors du calcul des futures solutions.

Cet article a pour but de donner une première introduction à la méthode des bases réduites.

Dans la section 2 nous donnerons les définitions, les notations et les résultats de bases utiles à la théorie. Ensuite dans les sections 3 à 6, nous donnerons un résumé de la méthode pour donner une idée au lecteur de son fonctionnement. Les sections 7 à 9 donnerons des résultats théoriques permettant de fournir une borne pour l'erreur a posteriori. Finalement dans la section 10, nous mettrons en pratique la méthode des bases réduites à l'aide d'un logiciel développé au MIT.

2 Résultats d'analyse fonctionnelle

Commençons par donner quelques définitions et résultats classiques d'analyse fonctionnelle. Ces derniers nous seront utiles par la suite pour traiter les différents problèmes.

2.1 Définition (Espace de Sobolev). *On considère l'espace $L^2(\Omega)$ défini par*

$$L^2(\Omega) = \left\{ f : \Omega \rightarrow \mathbb{R} \text{ mesurable au sens de Lebesgue} \left| \int_{\Omega} |f(x)|^2 dx < \infty \right. \right\}.$$

On désigne par $H^k(\Omega)$ l'espace de Sobolev d'ordre k défini par

$$H^k(\Omega) = \{f \in L^2(\Omega) \mid D^\alpha f \in L^2(\Omega), |\alpha| \leq k\}$$

où $D^\alpha f$ est la dérivée de f prise au sens des distributions. De plus on notera

$$H_0^k(\Omega) = \{f \in H^k(\Omega) \mid f = 0 \text{ sur } \partial\Omega\}$$

$$H_D^1 = \{f \in H^1 \mid f = 0 \text{ sur } \Gamma_D, D \subset \partial\Omega, D \text{ de mesure non nulle}\}.$$

2.2 Théorème (Théorème de représentation de Riesz). *Soit H un espace de Hilbert. Alors pour tout $F \in H'$, il existe un unique $v \in H$ tel que*

$$F(x) = (v, x)_H \quad \forall x \in H.$$

De plus $\|F\|_{H'} = \|v\|_H$.

3 Problématique

3.1 Définition du problème

Nous allons maintenant définir le problème modèle autour duquel nous développerons la théorie de base concernant la méthode des bases réduites. Considérons $\mathcal{D} \subset \mathbb{R}^p$ un domaine pour un certain paramètre et $\Omega \subset \mathbb{R}^d$ un domaine borné (avec p et d sont deux entiers). On définit de plus X^e l'espace des fonctions telles que pour $\nu \in \mathbb{N}$ on ait

$$(H_0^1(\Omega))^\nu \subset X^e \subset (H^1(\Omega))^\nu$$

Soient $\forall \mu \in \mathcal{D}$ une forme bilinéaire et continue $a(\cdot, \cdot; \mu)$ et deux fonctionnelles linéaires bornées $f(\cdot; \mu)$ et $l(\cdot; \mu)$. On se pose alors le problème suivant

$$\begin{cases} \text{Pour } \mu \in \mathcal{D}, \text{ évaluer} \\ s^e(\mu) = l(u^e(\mu); \mu) & \text{où } u^e(\mu) \in X^e \text{ satisfait} \\ a(u^e, v; \mu) = f(v; \mu) & \forall v \in X^e. \end{cases} \quad (1)$$

On introduit le produit scalaire suivant sur X^e :

$$\langle w, v \rangle_\mu \equiv a(w, v; \mu) \quad \forall w, v \in X^e. \quad (2)$$

Avec ce dernier, on définit une norme sur X^e :

$$\|w\|_\mu \equiv \langle w, w \rangle_\mu^{\frac{1}{2}} \quad \forall w \in X^e \quad (3)$$

Pour nos besoins, nous allons définir un second produit scalaire (lié à l'énergie du système) et la norme qui lui correspond :

$$(w, v)_X \equiv \langle w, v \rangle_\mu + \tau(w, v)_{L^2(\omega)} \quad \forall w, v \in X^e. \quad (4)$$

$$\|w\|_X \equiv (w, w)_X^{\frac{1}{2}} \quad \forall w \in X^e \quad (5)$$

Ceci nous permet de définir la constante de coercivité

$$\alpha^e(\mu) \equiv \inf_{w \in X^e} \frac{a(w, w; \mu)}{\|w\|_X^2} \quad (6)$$

et la constante de continuité

$$\gamma^e(\mu) = \sup_{w \in X^e} \sup_{v \in X^e} \frac{a(w, v; \mu)}{\|w\|_X \|v\|_X} \quad \forall \mu \in \mathcal{D}. \quad (7)$$

3.2 Restriction sur $a(\cdot, \cdot; \mu)$

Nous allons faire l'hypothèse cruciale que a est telle que

$$a(w, v; \mu) = \sum_{q=1}^Q \Theta^q(\mu) a^q(w, v) \quad (8)$$

où pour $q = 1, \dots, Q$ dépend de μ et $\Theta^q : \mathcal{C} \rightarrow \mathbb{R}$ et $a^q : X^e \times X^e \rightarrow \mathbb{R}$ est indépendante de μ .

Cette hypothèse restrictive sur a va nous permettre d'augmenter significativement les performances des calculs numériques en évitant de recalculer à chaque fois la forme bilinéaire $a(w, v; \mu)$. En effet, une fois les valeurs de $a_q(w, v)$ stockées, il suffit d'évaluer l'expression (8) pour obtenir la valeur de $a(w, v; \mu)$.

3.3 Approximation par éléments finis

Pour que nous puissions traiter notre problème numériquement, on approche X^e par une suite $X^{\mathcal{N}} \subset X^e$ d'approximation par Eléments finis (où $\dim(X^{\mathcal{N}}) = \mathcal{N}$). On obtient ainsi un nouveau problème :

$$\begin{cases} \text{Pour } \mu \in \mathcal{D}, \text{ évaluer} \\ s^{\mathcal{N}}(\mu) = f(u^{\mathcal{N}}(\mu)) & \text{où } u^{\mathcal{N}}(\mu) \in X^{\mathcal{N}} \text{ satisfait} \\ a(u^{\mathcal{N}}, v; \mu) = f(v; \mu) & \forall v \in X^{\mathcal{N}} \end{cases} \quad (9)$$

Comme avant on peut définir la constante de coercivité

$$\alpha^{\mathcal{N}}(\mu) \equiv \inf_{w \in X^{\mathcal{N}}} \frac{a(w, w; \mu)}{\|w\|_X^2} \quad (10)$$

et celle de continuité

$$\gamma^{\mathcal{N}}(\mu) = \sup_{w \in X^{\mathcal{N}}} \sup_{v \in X^{\mathcal{N}}} \frac{a(w, v; \mu)}{\|w\|_X \|v\|_X} \quad \forall \mu \in \mathcal{D}. \quad (11)$$

Ainsi il est clair que

$$\alpha^e(\mu) \geq \alpha_0 > 0 \quad \forall \mu \in \mathcal{D} \text{ et}$$

$$\alpha^{\mathcal{N}}(\mu) \geq \alpha^e(\mu) \quad \forall \mu \in \mathcal{D}.$$

où α_0 est la constante de coercivité de $a(\cdot, \cdot)$ et

$$\gamma^{\mathcal{N}}(\mu) \leq \gamma^e(\mu), \quad \forall \mu \in \mathcal{D}.$$

4 Représentations géométriques affines

La technique des bases réduites nécessite que Ω soit un domaine sans paramètre.

Si l'on veut considérer des combinaisons linéaires du domaine à différents instants, ces instants doivent être définis par rapport à une configuration du domaine Ω commune. Donc pour permettre des variations géométriques du domaine, on doit interpréter Ω (sans paramètre) comme la préimage de $\Omega_0(\mu)$ (dépendant d'un paramètre) via une application $\mathcal{T} : \Omega \rightarrow \Omega_0$. On appelle Ω le domaine de référence et $\Omega_0(\mu)$ le domaine actuel (ou original).

On peut donc définir à partir de $a_0(w_0, v_0; \mu)$ la forme

$$a(w, v; \mu) = a_0(\mathcal{T}(w), \mathcal{T}(v); \mu)$$

telle que grâce à (8)

$$a(w, v; \mu) = \sum_{q=1}^Q \Theta^q(\mu) a^q(w, v).$$

5 Considération sur les bases réduites

Considérons un système d'équations non-linéaires provenant d'un problème discrétisé dépendant d'un paramètre $\mu \in \mathcal{D}$. Un tel système peut s'écrire sous la forme

$$\mathcal{F}(u, \mu) = 0 \quad (12)$$

pour $u \in X^{\mathcal{N}}$, $\mathcal{F} : X^{\mathcal{N}} \times \mathcal{D} \rightarrow X^{\mathcal{N}}$ et $X^{\mathcal{N}}$ un ensemble de dimension \mathcal{N} finie. Soit $X_k^{\mathcal{N}}$ l'espace de la base réduite de dimension k satisfaisant

$$X_k^{\mathcal{N}} \subset X^{\mathcal{N}}.$$

Il existe plusieurs façons classiques de former des bases réduites, cependant nous allons nous concentrer ici sur les bases réduites de Taylor et de Lagrange. Cependant il est possible de produire les bases réduites de Hermite et POD, mais nous ne les traiterons pas dans ce document.

En fait la méthode des bases réduites est une méthode de réduction dont le but est de diminuer le nombre de degrés de liberté du problème.

5.1 Base réduite de Taylor

Supposons que nous possédons une valeur μ_0 de μ pour laquelle nous connaissons une solution u_0 pour le problème 12. On définit $u_k = \frac{\partial^k u}{\partial \mu^k}$ pour $k = 1, \dots, p$ avec $p \in \mathbb{N}$. u_k est obtenu en calculant la différentielle de (12) par rapport au paramètre puis en l'évaluant en (u_0, μ_0) . Par exemple si l'on souhaite calculer u_1 , il faut d'abord effectuer

$$\frac{d\mathcal{F}}{d\mu}(u_0, \mu_0) = \mathcal{F}_u(u_0, \mu_0)u_1 + \mathcal{F}_\mu(u_0, \mu_0) = 0.$$

On optient alors u_1 en résolvant le système

$$\mathcal{F}_u(u_0, \mu_0)u_1 = -\mathcal{F}_\mu(u_0, \mu_0).$$

Plus généralement pour obtenir u_k avec $k = 1, \dots, p$, il faut résoudre un système d'équations de la forme

$$\mathcal{F}_u(u_0, \mu_0)u_k = \mathcal{G}_k(u_0, u_1, \dots, u_{k-1}, \mu_0), \quad (13)$$

pour une certaine fonction \mathcal{G}_k . L'espace $X_{p+1}^{\mathcal{N}}$ peut alors être défini par

$$X_{p+1}^{\mathcal{N}} = \text{vect}\{u_0, u_1, \dots, u_p\}.$$

Remarque

L'optention de u_0 requière la résolution du système non-linéaire (12). Par contre l'optention de u_k avec $k = 1, \dots, p$ ne dépend que d'un système linéaire. De plus les coefficients de la matrice restent les mêmes. Ainsi le calcul des u_k avec $k = 2, \dots, p$ ne demande que du « backsolving ».

5.2 Base réduite de Lagrange

Supposons que nous connaissions u_0, \dots, u_p des solutions de (12) associées à différentes valeurs de μ que nous noterons μ_0, \dots, μ_p . Ces vecteurs constituent alors la base réduite de Lagrange. L'espace $X_{p+1}^{\mathcal{N}}$ peut alors être défini par

$$X_{p+1}^{\mathcal{N}} = \text{vect}\{u_0, u_1, \dots, u_p\}.$$

Cette méthode requière donc la résolution du système non-linéaire (12) afin de trouver $p + 1$ solutions.

Une fois que les vecteurs ont été trouvés, le processus de mise à jour peut être accompli un vecteur à la fois plutôt que de régénérer $p + 1$ vecteurs.

5.3 Généralisation

Le lecteur pourra trouver dans [6] une approche plus générale (utilisant les variétés) pour construire les bases réduites de Taylor et de Lagrange.

6 Méthode des bases réduites

6.1 Espace des bases réduites

Reprenons l'approximation $X^{\mathcal{N}}$ de l'espace par éléments finis de dimension \mathcal{N} que nous avons définie à la section 3.3. Soit $N_{max} \in \mathbb{N}$, on introduit la suite $X_1^{\mathcal{N}}, \dots, X_{N_{max}}^{\mathcal{N}}$ de sous-espaces de $X^{\mathcal{N}}$ (avec $\dim X_N^{\mathcal{N}} = N$). On suppose de plus que

$$X_1^{\mathcal{N}} \subset X_2^{\mathcal{N}} \subset \dots \subset X_{N_{max}}^{\mathcal{N}} \subset X^{\mathcal{N}} \quad (14)$$

Cette hypothèse, appelée « nested condition » ou « hierarchical condition », permet d'assurer une certaine efficacité mémoire lors du calcul numérique.

Pour la suite de cette section, nous allons nous intéresser aux bases réduites de Lagrange. Posons $W_N^{\mathcal{N}} = X_N^{\mathcal{N}}$ $1 \leq N \leq N_{max}$ pour spécifier que nous travaillons dans un espace des bases réduites de Lagrange de dimension N .

Pour commencer introduisons un ensemble principal de paramètres

$$\Xi = \{\mu^1, \dots, \mu^{N_{max}}\} \subset \mathcal{D}.$$

Ensuite pour $1 \leq N \leq N_{max}$, nous définissons les ensembles de paramètres de Lagrange

$$S_N = \{\mu^1, \dots, \mu^N\},$$

auxquels on associe l'espace des bases réduites correspondant

$$W_N^{\mathcal{N}} = \text{vect}\{u^{\mathcal{N}}(\mu^n), 1 \leq n \leq N\}.$$

On vérifie aisément que

$$S_1 \subset S_2 \subset \dots \subset S_{max}$$

et que les espaces de Lagrange satisfont la condition (14) :

$$W_1^{\mathcal{N}} \subset W_2^{\mathcal{N}} \subset \dots \subset W_{N_{max}}^{\mathcal{N}} \subset W^{\mathcal{N}}$$

Les $u^{\mathcal{N}}(\mu^n)$ pour $1 \leq n \leq N_{max}$ sont souvent appelés « snapshot ». Ce sont des solutions du problème pour certaines valeurs particulières du paramètre.

6.2 Algorithme glouton (« greedy ») pour la construction des espaces des bases réduites de Lagrange

Dans cette section, nous allons donner un algorithme permettant de construire les $W_N^{\mathcal{N}}$ pour $1 \leq N \leq N_{max}$ dont nous avons parlé dans la section 6.1.

Soit Ξ un ensemble fini de points de \mathcal{D} . Cette ensemble va nous servir de substitut pour \mathcal{D} lorsque nous chercherons à diminuer l'erreurs sur le domaine du paramètre. Nous appellerons Ξ *ensemble de test*.

Afin de pouvoir raffiner notre échantillon de paramètres, nous devons choisir Ξ suffisamment grand. Une façon de construire un tel ensemble est d'utiliser les méthodes de Monte-Carlo (avec la densité uniforme ou log-uniforme).

A présent, on cherche à construire les ensembles $W_N^{\mathcal{N} \text{ Greedy}}$. On se donne Ξ comme décrit auparavant, N_{max} et on définit $S_1 = \{\mu^1\}$, et $W_1^{\mathcal{N} \text{ Greedy}} = \text{span}\{u^{\mathcal{N}}(\mu^1)\}$. Ensuite pour $N = 2, \dots, N_{max}$, on va chercher

$$\mu^N = \arg \max_{\mu \in \Xi} \Delta_{N-1}(\mu)$$

où $\Delta_N = \Delta_N^{en}$ ou Δ_N^s sont respectivement les estimateurs des erreurs de la norme de l'énergie et de l'output $s_N^{\mathcal{N}}$. Ces deux quantités seront définies dans la section 9.1. En pratique, à la place de définir N_{max} , on peut se fixer une tolérance ε telle que l'on arrête de chercher de nouveau μ^N lorsque

$$\max_{\mu \in \Xi} \Delta_{N-1}(\mu) \leq \varepsilon.$$

Ensuite on peut définir par récurrence

$$S_N = S_{N-1} \cup \mu^N$$

et

$$W_N^{\mathcal{N} \text{ Greedy}} = W_{N-1}^{\mathcal{N} \text{ Greedy}} \cup \text{span}\{u^{\mathcal{N}}(\mu^N)\}.$$

6.3 Projection de Galerkin

Les projection de Galerkin offre approche efficace dans l'optique de notre travail. En effet nous obtenons le problème

$$\begin{cases} \text{Pour } \mu \in \mathcal{D}, \text{ évaluer} \\ s_N^{\mathcal{N}}(\mu) = f(u^{\mathcal{N}}(\mu)) & \text{où } u^{\mathcal{N}}(\mu) \in X_N^{\mathcal{N}} \subset X^{\mathcal{N}} \text{ satisfait} \\ a(u^{\mathcal{N}}, v; \mu) = f(v) & \forall v \in X_N^{\mathcal{N}}. \end{cases} \quad (15)$$

Notre objectif est avant tout d'obtenir une évaluation de $s_N^{\mathcal{N}}(\mu)$. Le lemme 6.3.1, va nous permettre de constater que les $s_N^{\mathcal{N}}$ vont converger vers les $s^{\mathcal{N}}$ de manière quadratique à mesure que les $u_N^{\mathcal{N}}$ vont se rapprocher de $u^{\mathcal{N}}$.

- 6.3.1 Lemme.**
1. $\|u^{\mathcal{N}}(\mu) - u_N^{\mathcal{N}}(\mu)\|_{\mu} \leq \inf_{w \in X_N^{\mathcal{N}}} \|u^{\mathcal{N}}(\mu) - w\|_{\mu}$
 2. $s^{\mathcal{N}}(\mu) - s_N^{\mathcal{N}}(\mu) = \|u^{\mathcal{N}}(\mu) - u_N^{\mathcal{N}}(\mu)\|_{\mu}^2$

Démonstration. Le premier résultat est obtenu de manière direct. Pour le second, on a

$$\begin{aligned} s^{\mathcal{N}} - s_N^{\mathcal{N}} &= f(u^{\mathcal{N}}) - f(u_N^{\mathcal{N}}) \\ &= a(u_N^{\mathcal{N}}, u^{\mathcal{N}}) - a(u_N^{\mathcal{N}}, u_N^{\mathcal{N}}) \\ &= a(u_N^{\mathcal{N}}, u^{\mathcal{N}} - u_N^{\mathcal{N}}) \end{aligned}$$

Or $a(u_N^{\mathcal{N}}, u^{\mathcal{N}} - u_N^{\mathcal{N}}) = a(u^{\mathcal{N}} - u_N^{\mathcal{N}}, u^{\mathcal{N}} - u_N^{\mathcal{N}})$ par l'orthogonalité de Galerkin. Et donc

$$s^{\mathcal{N}} - s_N^{\mathcal{N}} = a(u^{\mathcal{N}} - u_N^{\mathcal{N}}, u^{\mathcal{N}} - u_N^{\mathcal{N}}) = \|u^{\mathcal{N}}(\mu) - u_N^{\mathcal{N}}(\mu)\|_{\mu}^2.$$

□

Nous allons maintenant nous intéresser à l'approximation de Galerkin du problème énoncée en (15). La première chose à faire est de trouver une bonne base pour notre espace, faute de quoi, nous risquons d'obtenir un système mal conditionné. Pour y arriver nous devons utiliser le procédé de Gram-Schmidt avec le produit scalaire $(\cdot, \cdot)_X$ à nos *snapshots* $u^{\mathcal{N}}(\mu^n)$, $1 \leq n \leq N_{max}$ afin d'obtenir des fonctions de base $\zeta_n^{\mathcal{N}}$, $1 \leq n \leq N_{max}$. Il nous suffit ensuite de choisir l'ensemble $\{\zeta_n^{\mathcal{N}}\}_{n=1, \dots, N}$ comme pour pour $W_N^{\mathcal{N}}$, $1 \leq N \leq N_{max}$.

Nous pouvons à présent réécrire $u_N^{\mathcal{N}}$ à l'aide de cette base :

$$u_N^{\mathcal{N}}(\mu) = \sum_{m=1}^N u_{Nm}^{\mathcal{N}}(\mu) \zeta_m^{\mathcal{N}}. \quad (16)$$

En posant $v = \zeta_n^{\mathcal{N}}$, $1 \leq n \leq N$ dans (15), on obtient les équations de « raideurs »

$$\sum_{m=1}^N a(\zeta_m^{\mathcal{N}}, \zeta_n^{\mathcal{N}}; \mu) u_{Nm}^{\mathcal{N}}(\mu) = f(\zeta_n^{\mathcal{N}}) \quad 1 \leq n \leq N, \quad (17)$$

de plus nous pouvons évaluer notre valeur de sortie comme

$$s_N^{\mathcal{N}} = \sum_{m=1}^N u_{Nm}^{\mathcal{N}}(\mu) f(\zeta_m^{\mathcal{N}}).$$

6.4 Procédure Online-Offline

Un système d'équation tel que (17) ne pose aucun problème lors de sa résolution. Par contre dès que l'on change la valeur de notre paramètre μ , il faut alors recalculer un nouveau système et le résoudre. Ainsi l'évaluation

$$\mu \rightarrow s_N(\mu)$$

va nécessiter un coût inacceptable. Heureusement l'hypothèse (8) nous permet de réécrire le système sous la forme

$$\sum_{m=1}^N \left(\sum_{q=1}^Q \Theta^q(\mu) a^q(\zeta_m^{\mathcal{N}}, \zeta_n^{\mathcal{N}}) \right) u_{Nm}^{\mathcal{N}}(\mu) = f(\zeta_n^{\mathcal{N}}) \quad 1 \leq n \leq N. \quad (18)$$

Remarquons que puisque les $\zeta^{\mathcal{N}}$ ne dépendent pas de μ , il est possible de pré-calculer

$$f(\zeta_n^{\mathcal{N}}), \quad 1 \leq n \leq N_{max} \quad (19)$$

et

$$a^q(\zeta_m^{\mathcal{N}}, \zeta_n^{\mathcal{N}}), \quad 1 \leq n \leq N_{max}, 1 \leq q \leq Q. \quad (20)$$

Ces calculs constituent la partie « offline » de la procédure et l'évaluation de (18) en μ la partie « online ».

7 Borne inférieure pour la constante de coercivité

Nous allons nous intéresser à une démarche qui permet de construire une borne inférieure pour la constante de coercivité et dans le cas non-coercif la constante pour la stabilité inf-sup. La méthode que nous allons décrire est appelée *Méthode des Contraintes Successives* (SCM en anglais). Cette dernière réduit la partie « online » à la résolution d'un programme linéaire dont le nombre d'opérations est indépendant de \mathcal{N} .

Comme nous le verrons à la section 9, cette borne est importante lorsque l'on fera l'analyse de l'erreur a priori pour les approximations des bases réduites paramétrant des équations aux dérivées partielles.

7.1 Méthode des Contraintes successives

On rappelle la définition de la constante de coercivité pour les éléments finis

$$\alpha^{\mathcal{N}}(\mu) = \inf_{w \in X^{\mathcal{N}}} \frac{a(w, w; \mu)}{\|w\|_X^2}.$$

On suppose que $a(., .)$ est continue et coercive avec constante $\alpha^{\mathcal{N}}(\mu) > 0 \forall \mu \in \mathcal{D}$. On rappelle également l'hypothèse (8).

On introduit une *fonction objectif*

$$\begin{aligned} \mathcal{J}^{obj} : \mathcal{D} \times \mathbb{R}^Q &\rightarrow \mathbb{R} \\ (\mu, y) &\mapsto \sum_{q=1}^Q \Theta^q(\mu) y_q \end{aligned}$$

où $y = (y_1, \dots, y_Q)$.

7.1.1 Lemme. *La constante de coercivité peut être exprimée comme*

$$\alpha^{\mathcal{N}}(\mu) = \inf_{y \in \mathcal{Y}} \mathcal{J}^{obj}(\mu; y)$$

où

$$\mathcal{Y} = \left\{ y \in \mathbb{R}^Q \mid \exists w_y \in X^{\mathcal{N}} \text{ tel que } y_q = \frac{a(w_y, w_y)}{\|w_y\|_X^2}, 1 \leq q \leq Q \right\}.$$

Démonstration. Soit $\mu \in \mathcal{D}$.

$$\begin{aligned}
 \alpha^{\mathcal{N}}(\mu) &= \inf_{w \in X^{\mathcal{N}}} \frac{a(w, w; \mu)}{\|w\|_X^2} \\
 &= \inf_{w \in X^{\mathcal{N}}} \frac{\sum_{q=1}^Q \Theta^q(\mu) a^q(w, w)}{\|w\|_X^2} \quad \text{par (8)} \\
 &= \inf_{w \in X^{\mathcal{N}}} \sum_{q=1}^Q \Theta^q(\mu) z(w) \quad \text{où } z(w) = \frac{a^q(w, w)}{\|w\|_X^2} \\
 &= \inf_{y \in \mathcal{Y}} \mathcal{J}^{obj}(\mu, y).
 \end{aligned}$$

□

Introduisons à présent la *boîte de contrainte de continuité*

$$\mathcal{B} = \prod_{q=1}^Q \left[\inf_{w \in X^{\mathcal{N}}} \frac{a^q(w, w)}{\|w\|_X^2}, \sup_{w \in X^{\mathcal{N}}} \frac{a^q(w, w)}{\|w\|_X^2} \right]. \quad (21)$$

On remarque que la continuité de $a(\cdot, \cdot)$ nous assure que \mathcal{B} est bornée.

On définit ensuite l'*échantillon de la contrainte de coercivité*

$$C_J = \{\mu_{SCM}^1 \in \mathcal{D}, \dots, \mu_{SCM}^J \in \mathcal{D}\}. \quad (22)$$

On note de plus $C_J^{M, \mu}$ l'ensemble des $M \geq 1$ points de C_J les plus proches d'un certain $\mu \in \mathcal{D}$. En particulier, si $M > J$, on pose $C_J^{M, \mu} = C_J$.

On verra à la section 8.4 comment former les C_J .

7.1.2 Définition. Soient C_J , $M \in \mathbb{N} = \{1, 2, \dots\}$ et $\mu \in \mathcal{D}$. On définit l'ensemble de la borne inférieure $\mathcal{Y}_{LB}(\mu; C_J, M)$ comme

$$\mathcal{Y}_{LB}(\mu; C_J, M) = \left\{ y \in \mathbb{R}^Q \mid y \in \mathcal{B}, \mathcal{J}^{obj}(\mu', y) \geq \alpha^{\mathcal{N}}(\mu'), \forall \mu' \in C_J^{M, \mu} \right\}.$$

7.1.3 Lemme. Soient $C_J \subset \mathcal{D}$ et $M \in \mathbb{N}$, alors

$$\mathcal{Y} \subset \mathcal{Y}_{LB}(\mu; C_J, M) \quad \forall \mu \in \mathcal{D}.$$

Démonstration. Soit $y \in \mathcal{Y}$. Alors par définition de \mathcal{Y} , il existe $w_y \in X^{\mathcal{N}}$ tel que

$$y_q = \frac{a^q(w_y, w_y)}{\|w_y\|_X^2} \quad 1 \leq q \leq Q.$$

On remarque les deux éléments suivants :

1. $y_q \in \mathcal{B}$. En effet $\inf_{w \in X^{\mathcal{N}}} \frac{a^q(w, w)}{\|w\|_X^2} \leq y_q \leq \sup_{w \in X^{\mathcal{N}}} \frac{a^q(w, w)}{\|w\|_X^2}$.
2. $\sum_{q=1}^Q \Theta^q(\mu) \frac{a^q(w_y, w_y)}{\|w_y\|_X^2} = \frac{a(w_y, w_y; \mu)}{\|w_y\|_X^2} \geq \alpha^{\mathcal{N}}(\mu) \quad \forall \mu \in \mathcal{D}$.

Il en découle directement que $y \in \mathcal{Y}_{LB}(\mu; C_J, M)$. □

7.1.4 Définition. On définit la borne inférieure par

$$\alpha_{LB}^{\mathcal{N}}(\mu; C_J, M) = \min_{y \in \mathcal{Y}_{LB}(\mu; C_J, M)} \mathcal{J}^{obj}(\mu, y).$$

7.1.5 Proposition. Soient $C_J \subset \mathcal{D}$, $M \in \mathbb{N}$, alors

$$\alpha_{LB}^{\mathcal{N}}(\mu) \leq \alpha^{\mathcal{N}}(\mu) \quad \forall \mu \in \mathcal{D}.$$

Démonstration. Soit $\mu \in \mathcal{D}$.

$$\begin{aligned} \alpha_{LB}^{\mathcal{N}}(\mu) &= \min_{y \in \mathcal{Y}_{LB}(\mu; C_J, M)} \mathcal{J}^{obj}(\mu, y) && \text{par définition} \\ &\leq \min_{y \in \mathcal{Y}} \mathcal{J}^{obj}(\mu, y) && \text{par le lemme 7.1.3} \\ &= \alpha^{\mathcal{N}}(\mu). \end{aligned}$$

□

7.2 Discussion

La recherche de la borne inférieure $\alpha_{LB}^{\mathcal{N}}$ est un problème linéaire de minimisation. Il contient les Q variables

$$q_1, \dots, q_Q$$

qui sont soumises à $2Q + M$ contraintes d'inégalités :

- Chaque y_i est soumis à deux contraintes venant de la définition de \mathcal{B} (21).
 - $\mathcal{J}^{obj}(\mu', y) \geq \alpha^{\mathcal{N}}(\mu')$, $\forall \mu' \in C_J^{M, \mu}$ et $|C_J^{M, \mu}| \leq M$.
- De plus, on observe que, connaissant \mathcal{B} et l'ensemble

$$\{\alpha^{\mathcal{N}}(\mu') \mid \mu' \in C_J\},$$

le coût de l'évaluation $\mu \rightarrow \alpha_{LB}^{\mathcal{N}}(\mu)$ est indépendant de \mathcal{N} .

On reviendra sur le détail dans la section 9.3.

8 Borne supérieure pour la constante de coercivité

Dans cette section nous allons nous intéresser au calcul d'une borne supérieure pour la constante de coercivité. Cette borne va permettre la construction efficace d'un échantillon de la contrainte de coercivité C_J (22).

Soient C_J , $M \in \mathbb{N}$ et $\mu \in \mathcal{D}$. On définit

$$y^*(\mu) = \arg \inf_{y \in \mathcal{Y}} \mathcal{J}^{obj}(\mu; y)$$

afin d'introduire l'ensemble de la borne supérieure :

$$\mathcal{Y}_{UB}(\mu; C_J, M) = \{y^*(\mu') \mid \mu' \in C_J^{M, \mu}\}.$$

Cette ensemble nous permet d'introduire notre borne supérieure :

$$\alpha_{UB}^{\mathcal{N}}(\mu; C_J, M) = \min_{y \in \mathcal{Y}_{UB}(\mu; C_J, M)} \mathcal{J}^{obj}(\mu; y). \quad (23)$$

8.1 Lemme. Soient C_J et $M \in \mathbb{N}$, alors

$$\mathcal{Y}_{UB}(\mu; C_J, M) \subset \mathcal{Y} \quad \forall \mu \in \mathcal{D}$$

Démonstration. Soit $y \in \mathcal{Y}_{UB}(\mu; C_J, M)$. Donc

$$\begin{aligned} y(\mu) &= \arg \inf_{\tilde{y} \in \mathcal{Y}} \mathcal{J}^{obj}(\mu, \tilde{y}) \\ &= \tilde{y} \in \mathcal{Y}. \end{aligned}$$

Ainsi $\mathcal{Y}_{UB}(\mu; C_J, M) \subset \mathcal{Y}$. \square

8.2 Corollaire. Soient C_J et $M \in \mathbb{N}$, alors $\alpha_{UB}^{\mathcal{N}}(\mu; C_J, M) \geq \alpha^{\mathcal{N}}(\mu) \forall \mu \in \mathcal{D}$.

Démonstration. Le résultat découle immédiatement des lemmes 7.1.1 et 8.1. \square

8.3 Discussion

La borne supérieure (23) est obtenue en évaluant $\mu \rightarrow \alpha_{UB}^{\mathcal{N}}(\mu)$. Si on suppose que l'on connaît l'ensemble

$$\{y^*(\mu') \mid \mu' \in C_J\}$$

et donc $\mathcal{Y}_{UB}(\mu; C_J, M)$, alors le nombre d'opérations pour la déterminer est indépendant de \mathcal{N} .

8.4 Algorithme glouton de sélection des C_J

Nous allons à présent décrire une procédure « offline » permettant de construire les C_J . On se donne pour commencer un échantillon d'« entraînement » contenant $n_{train,SCM}$ points de notre ensemble de paramètres \mathcal{D} :

$$\Xi_{train,SCM} = \{\mu_{train,SCM}^1, \dots, \mu_{train,SCM}^{n_{train,SCM}}\} \subset \mathcal{D}$$

Ensuite on se fixe une tolérance ε_{SCM} . Ensuite on utilise l'algorithme 1 afin de construire nos C_J .

Initialisation
 $J = 1$
 $C_1 = \{\mu_{SCM}^1\}$ [μ_{SCM}^1 est choisi arbitrairement]

Tant que ($\max_{\mu \in \Xi_{train,SCM}} \left[\frac{\alpha_{UB}^{\mathcal{N}}(\mu; C_J, M) - \alpha_{LB}^{\mathcal{N}}(\mu; C_J, M)}{\alpha_{UB}^{\mathcal{N}}(\mu; C_J, M)} \right] > \varepsilon_{SCM}$) **faire**

1. $\mu_{SCM}^{J+1} = \arg \max_{\mu \in \Xi_{train,SCM}} \left[\frac{\alpha_{UB}^{\mathcal{N}}(\mu; C_J, M) - \alpha_{LB}^{\mathcal{N}}(\mu; C_J, M)}{\alpha_{UB}^{\mathcal{N}}(\mu; C_J, M)} \right]$

2. $C_{J+1} = C_J \cup \mu_{SCM}^{J+1}$

3. $J \leftarrow J + 1$

Fait
Poser $J_{max} = J$

Algorithme 1: Algorithme glouton de sélection des C_J

En fait l'algorithme choisit successivement les éléments dans $\Xi_{train,SCM}$ pour lesquels la différence entre $\alpha_{LB}^{\mathcal{N}}(\mu, C_J, M)$ et $\alpha_{UB}^{\mathcal{N}}(\mu, C_J, M)$ est la plus grande. Ainsi après chaque itération, cette différence est de plus en plus petite. L'algorithme s'arrête lorsque cette différence est inférieure à la tolérance ε_{SCM} .

De cette algorithme, on déduit le lemme suivant

8.4.1 Lemme. *Soient $M \in \mathbb{N}$ et $C_{J_{max}}$ donné par l'algorithme 1. Alors*

$$\frac{\alpha^{\mathcal{N}}(\mu)}{\alpha_{LB}^{\mathcal{N}}(\mu; C_{J_{max}}, M)} \leq \frac{1}{1 - \varepsilon_{SCM}}, \quad \forall \mu \in \Xi_{train,SCM}$$

Démonstration.

$$\begin{aligned} \frac{\alpha^{\mathcal{N}}(\mu)}{\alpha_{LB}^{\mathcal{N}}(\mu; C_{J_{max}}, M)} &= \frac{\alpha^{\mathcal{N}}(\mu)}{\alpha_{UB}^{\mathcal{N}}(\mu; C_{J_{max}}, M) - (\alpha_{UB}^{\mathcal{N}}(\mu, C_{J_{max}}, M) - \alpha_{LB}^{\mathcal{N}}(\mu; C_{J_{max}}, M))} \\ &= \frac{\alpha^{\mathcal{N}}(\mu)}{\alpha_{UB}^{\mathcal{N}}(\mu; C_{J_{max}}, M) \left(1 - \frac{\alpha_{UB}^{\mathcal{N}}(\mu, C_{J_{max}}, M) - \alpha_{LB}^{\mathcal{N}}(\mu; C_{J_{max}}, M)}{\alpha_{UB}^{\mathcal{N}}(\mu; C_{J_{max}}, M)}\right)} \\ &\leq \frac{\alpha^{\mathcal{N}}(\mu)}{\alpha_{UB}^{\mathcal{N}}(\mu; C_{J_{max}}, M)} \frac{1}{1 - \varepsilon_{SCM}} \\ &\leq \frac{1}{1 - \varepsilon_{SCM}}, \quad \forall \mu \in \Xi_{train,SCM} \end{aligned}$$

□

8.5 Procédure de calcul « offline-online »

Nous allons à présent nous représenter les coûts de calcul nécessité par les parties « online » et « offline ».

8.5.1 Partie « offline »

Pour cette partie on considère les opérations suivantes :

1. Résolution de $2Q$ problèmes de valeurs propres sur $X^{\mathcal{N}}$ afin de construire \mathcal{B} .
2. Résolution de J_{max} problèmes de valeurs propres sur $X^{\mathcal{N}}$ afin de construire $\{\alpha^{\mathcal{N}}(\mu') \mid \mu' \in C_{J_{max}}\}$.
3. Le produit intérieur $J_{max}Q$ sur $X^{\mathcal{N}}$ pour construire $\{y^*(\mu') \mid \mu' \in C_{J_{max}}\}$.
4. $n_{train,SCM}J_{max}$ borne inférieure de problème linéaire de taille $2Q + M$ pour obtenir $\arg \max$.

Le coût total dépend de \mathcal{N} et est de l'ordre de

$$O(\mathcal{N}(2Q + J_{max})) + O(\mathcal{N}QJ_{max}) + O(n_{train,SCM}J_{max}QM).$$

Par contre, aucun terme de la forme $O(n_{train,SCM}\mathcal{N})$ n'est présent. Ainsi il est possible de choisir n et \mathcal{N} grand sans risquer de faire enfler les calculs.

8.5.2 Partie « online »

Pour chaque évaluation $\mu \rightarrow \alpha_{LB}^{\mathcal{N}}(\mu; C_J, M)$ il faut

1. Déterminer l'ensemble $C_{J_{max}}^{M, \mu}$. Pour cela il faut ordonner les J_{max} points dans $C_{J_{max}}$. Cette opération coûte au plus $O(MJ_{max})$.
2. On doit ensuite procéder à $(M + 1)Q$ évaluations :

$$\mu^q \rightarrow \Theta^q(\mu).$$

Ceci implique donc un coût de $O((M + 1)Q)$.

3. Extraction de M membres de l'ensemble $\{\alpha^{\mathcal{N}}(\mu') \mid \mu' \in C_J\}$ que nous avons calculé dans la partie « offline ». Puis résolution d'un problème linéaire linéaire pour obtenir $\alpha_{LB}^{\mathcal{N}}(\mu; C_J, M)$.

On note que l'évaluation de $\alpha_{LB}^{\mathcal{N}}(\mu; C_J, M)$ ne dépend pas de \mathcal{N} .

9 Estimation de l'erreur a posteriori

Nous allons à présent nous intéresser l'erreur a posteriori. On souhaite en effet maîtriser la quantité $s_{\mathcal{N}}^{\mathcal{N}}(\mu)$ et donc pouvoir assurer la fiabilité et l'efficacité de l'approximation par les bases réduites. De plus comme nous l'avons déjà vu, elle joue un rôle dans l'algorithme glouton.

On rappelle que l'on a considéré le problème éléments finis (9). Ensuite nous avons utilisé les projections de Galerkin pour obtenir le problème (15).

On définit

$$e(\mu) \equiv u^{\mathcal{N}}(\mu) - u_{\mathcal{N}}^{\mathcal{N}}(\mu) \in X^{\mathcal{N}}.$$

La linéarité de $a(., .)$ nous donne

$$\begin{aligned} a(e(\mu), v; \mu) &= a(u^{\mathcal{N}}(\mu), v; \mu) - a(u_{\mathcal{N}}^{\mathcal{N}}(\mu), v; \mu) \\ &= f(v; \mu) - a(u_{\mathcal{N}}^{\mathcal{N}}(\mu), v; \mu) \quad \forall v \in X^{\mathcal{N}}. \end{aligned}$$

En posant

$$r(v; \mu) = f(v; \mu) - a(u_{\mathcal{N}}^{\mathcal{N}}(\mu), v; \mu), \quad (24)$$

on obtient l'équation du résidu :

$$a(e(\mu), v; \mu) = r(v; \mu) \quad (25)$$

Pour des raisons pratiques, nous allons faire intervenir la représentation de Riesz pour $r(v; \mu)$. Par le théorème de représentation de Riesz (2.2), on sait qu'il existe une unique fonction $\hat{e}(\mu)$ telle que

$$r(v; \mu) = (\hat{e}(\mu), v)_X \quad \forall v \in X^{\mathcal{N}}. \quad (26)$$

Ainsi il découle immédiatement de (25) que $a(e(\mu), v; \mu) = (\hat{e}(\mu), v)_X$. Le théorème de Riesz nous permet également de déterminer que

$$\|r(., \mu)\|_{(X^{\mathcal{N}})'} \equiv \frac{r(v; \mu)}{\|v\|_X} = \|\hat{e}(\mu)\|_X. \quad (27)$$

9.1 Borne de l'erreur

Nous allons à présent donner une borne pour notre erreur. A l'aide de la borne inférieure $\alpha_{LB}^{\mathcal{N}}(\mu)$ que nous avons construit dans la section 7, on définit l'estimateur de l'erreur pour la norme de l'énergie

$$\Delta_N^{en}(\mu) = \frac{\|\hat{e}(\mu)\|_X}{\sqrt{\alpha_{LB}^{\mathcal{N}}(\mu)}} \quad (28)$$

et l'estimateur de l'erreur pour l'output

$$\Delta_N^s(\mu) = \frac{\|\hat{e}(\mu)\|_X^2}{\alpha_{LB}^{\mathcal{N}}(\mu)}. \quad (29)$$

Enfin on introduit l'efficacité correspondant à chacun de ces estimateurs :

$$\eta_N^{en}(\mu) = \frac{\Delta_N^{en}(\mu)}{\| |w^{\mathcal{N}}(\mu) - w_N^{\mathcal{N}}(\mu) | \|_{\mu}} \quad (30)$$

$$\eta_N^s(\mu) = \frac{\Delta_N^s(\mu)}{s^{\mathcal{N}}(\mu) - s_N^{\mathcal{N}}(\mu)}. \quad (31)$$

où $\| |w | \|_{\mu} = \sqrt{a(w, w; \mu)}$

Remarque

Les efficacités introduites auparavant sont des outils pour mesurer la qualité des estimateurs que l'on a définis. En fait plus l'efficacité sera proche de 1, plus l'erreur sera faible.

9.1.1 Proposition. *Pour tous $N = 1, \dots, N_{max}$ et pour tous $\mu \in \mathcal{D}$, l'efficacité satisfait*

1. $1 \leq \eta_N^{en} \leq \sqrt{\frac{\gamma^e(\mu)}{\alpha_{LB}^{\mathcal{N}}(\mu)}}$,
2. $1 \leq \eta_N^s \leq \frac{\gamma^e(\mu)}{\alpha_{LB}^{\mathcal{N}}(\mu)}$.

Démonstration. Remarquons que l'on déduit immédiatement des définitions que les efficacités η_N^{en} et η_N^s sont supérieures ou égales à 1. On sait que $a(e(\mu), v; \mu) = (\hat{e}(\mu), v)_x \forall v \in X^{\mathcal{N}}$.

1. Choisissons $v = e(\mu)$,

$$\begin{aligned} a(e(\mu), e(\mu)) &= \| |e(\mu) | \|_{\mu}^2 \\ &= (\hat{e}(\mu), e(\mu))_X^2 \\ &\leq \|\hat{e}(\mu)\|_X \|e(\mu)\|_X \text{ par Cauchy-Schwarz.} \end{aligned}$$

Mais

$$\begin{aligned} \| |e(\mu) | \|_{\mu} &\equiv \sqrt{a(e(\mu), e(\mu))} \frac{\|\hat{e}(\mu)\|_X}{\|\hat{e}(\mu)\|_X} \\ &\geq \sqrt{\alpha^{\mathcal{N}}} \|e(\mu)\|_X. \end{aligned}$$

Ainsi

$$\|e(\mu)\|_\mu^2 \leq \|\hat{e}(\mu)\|_X \frac{\|e(\mu)\|_\mu}{\sqrt{\alpha^N}}.$$

On obtient finalement le résultat

$$\|e(\mu)\|_\mu \leq \Delta_N^{en}(\mu).$$

2. Choisissons $v = \hat{e}(\mu)$.

$$\begin{aligned} (\hat{e}(\mu), \hat{e}(\mu))_X &= \|\hat{e}(\mu)\|_X^2 \\ &= a(e(\mu), \hat{e}(\mu); \mu) \\ &\leq \|e(\mu)\|_\mu \|\hat{e}(\mu)\|_\mu \text{ par Cauchy-Schwarz.} \end{aligned}$$

Or

$$\begin{aligned} \|\hat{e}(\mu)\|_\mu &= \sqrt{a(\hat{e}(\mu), \hat{e}(\mu))} \frac{\|\hat{e}(\mu)\|_X}{\|\hat{e}(\mu)\|_X} \\ &\leq \sqrt{\gamma^e(\mu)} \|\hat{e}(\mu)\|_X. \end{aligned}$$

Ainsi

$$\|\hat{e}(\mu)\|_X^2 \leq \sqrt{\gamma^e(\mu)} \|\hat{e}(\mu)\|_X \|e(\mu)\|_\mu.$$

D'où

$$\begin{aligned} \Delta_N^{en} &\equiv \sqrt{\alpha_{LB}^N} \|\hat{e}(\mu)\|_X \\ &\leq \sqrt{\alpha_{LB}^N} \sqrt{\gamma^e(\mu)} \|e(\mu)\|_\mu. \end{aligned}$$

Ceci nous donne

$$\eta_N^{en} = \frac{\Delta_N^{en}}{\|e(\mu)\|_\mu} \leq \sqrt{\frac{\gamma^e(\mu)}{\alpha_{LB}^N}}. \quad (32)$$

On rappelle alors que $s^N(\mu) - s_N^N(\mu) = \|e(\mu)\|_\mu^2$ (voir lemme 6.3.1). Donc puisque $\Delta_N^s(\mu) = (\Delta_N^{en}(\mu))^2$, on a que

$$\begin{aligned} \eta_N^s(\mu) &\equiv \frac{\Delta_N^s(\mu)}{s^N(\mu) - s_N^N(\mu)} \\ &= \frac{(\Delta_N^{en}(\mu))^2}{\|e(\mu)\|_\mu^2} \text{ par le rappel ci-dessus,} \\ &= (\eta_N^{en}(\mu))^2 \\ &\leq \frac{\gamma^e(\mu)}{\alpha_{LB}^N} \text{ par (32).} \end{aligned}$$

□

On va maintenant donner un corollaire qui fait le lien entre l'output et Δ_N^s .

9.1.2 Corollaire. *Pour tous $N = 1, \dots, N_{max}$ et pour tous $\mu \in \mathcal{D}$, l'inégalité suivante est vérifiée :*

$$|s^N(\mu) - s_N^N(\mu)| \leq \Delta_N^s = (\Delta_N^{en})^2.$$

Démonstration. Par la définition de η_N^s , on obtient :

$$\begin{aligned}\Delta_N^s(\mu) &= \eta_N^s(\mu) |s^N(\mu) - s_N^N(\mu)| \\ &\geq |s^N(\mu) - s_N^N(\mu)| \quad \text{par la proposition 9.1.1}\end{aligned}$$

□

Revenons aux *efficacités* et montrons que η_N^{en} et η_N^s sont indépendantes de N et \mathcal{N} et ainsi qu'elles sont stables pour des raffinements éléments finis. C'est l'objet du théorème suivant :

9.1.3 Théorème (Stabilité des efficacités par des raffinements par éléments finis). *Pour un certain $\varepsilon_{SCM} \in (0, 1)$ fixé, il est possible de choisir*

$$\Xi = \{\mu_{SCM}^1, \dots, \mu_{SCM}^{N_{max}}\} \subset \mathcal{D}$$

tel que pour tous $N = 1, \dots, N_{max}$ et pour tous $\mu \in \Xi$, l'efficacité satisfait

1. $1 \leq \eta_N^{en} \leq \sqrt{\frac{1}{1-\varepsilon_{SCM}} \frac{\gamma^e(\mu)}{\alpha^e(\mu)}}$,
2. $1 \leq \eta_N^s \leq \frac{1}{1-\varepsilon_{SCM}} \frac{\gamma^e(\mu)}{\alpha^e(\mu)}$.

Démonstration.

$$\begin{aligned}\frac{\gamma^e(\mu)}{\alpha_{LB}^N(\mu)} &= \frac{\alpha^e(\mu)}{\alpha_{LB}^N(\mu)} \frac{\gamma^e(\mu)}{\alpha^e(\mu)} \quad \text{par la proposition 9.1.1} \\ &\leq \frac{\alpha^N(\mu)}{\alpha_{LB}^N(\mu)} \frac{\gamma^e(\mu)}{\alpha^e(\mu)} \\ &\leq \frac{1}{1-\varepsilon_{SCM}} \frac{\gamma^e(\mu)}{\alpha^e(\mu)} \quad \text{par le lemme 8.4.1}\end{aligned}$$

□

9.2 Approche pratique pour calculer l'erreur

Jusqu'à présent nous n'avons développé que des résultats théoriques. Cette section propose une méthode pour calculer $\|\hat{e}(\mu)\|_x$.

Pour y arriver, on utilise la définition de résidu (24) :

$$\begin{aligned}r(v; \mu) &\equiv f(v) - a(u_N^N(\mu), v; \mu) \\ &= f(v) - a\left(\sum_{n=1}^N u_N^N(\mu) \zeta_n^N, v; \mu\right) \quad \text{par (16)} \\ &= f(v) - \sum_{n=1}^N u_N^N(\mu) a(\zeta_n^N, v; \mu) \quad \text{par linéarité} \\ &= f(v) - \sum_{n=1}^N u_N^N(\mu) \sum_{q=1}^Q \Theta^q(\mu) a^q(\zeta_n^N, v) \quad \text{par (8)}\end{aligned}$$

Ainsi en utilisant (26), on obtient

$$\begin{aligned}(\hat{e}(\mu), v)_X &\equiv r(v; \mu) \\ &= f(v) - \sum_{n=1}^N \sum_{q=1}^Q \Theta^q(\mu) u_N^N(\mu) a^q(\zeta_n^N, v).\end{aligned}$$

De cette équation on déduit une expression pour $\hat{e}(\mu)$:

$$\hat{e}(\mu) = \mathcal{C} + \sum_{q=1}^Q \sum_{n=1}^N \Theta^q(\mu) u_{N_n}^{\mathcal{N}}(\mu) a^q(\zeta_n^{\mathcal{N}}, v)$$

où \mathcal{C} et \mathcal{L}_n^q , $1 \leq n \leq N$, $1 \leq q \leq Q$ sont les solutions éléments finis du problème de poisson suivant :

$$\begin{cases} (\mathcal{C}, v)_X &= f(v) \quad \forall v \in X^{\mathcal{N}} \\ (\mathcal{L}_n^q, v)_X &= -a^q(\zeta_n^{\mathcal{N}}, v) \quad \forall v \in X^{\mathcal{N}}. \end{cases} \quad (33)$$

On peut donc calculer $\|\hat{e}(\mu)\|_X$:

$$\begin{aligned} \|\hat{e}(\mu)\|_X^2 &= \left(\mathcal{C} + \sum_{q=1}^Q \sum_{n=1}^N \Theta^q(\mu) u_{N_n}^{\mathcal{N}}(\mu) a^q(\zeta_n^{\mathcal{N}}, v), \bullet \right)_X \\ &= (\mathcal{C}, \mathcal{C})_X + 2 \sum_{q=1}^Q \sum_{n=1}^N \Theta^q(\mu) u_{N_n}^{\mathcal{N}}(\mu) (\mathcal{C}, \mathcal{L}_n^q)_X \\ &\quad + \sum_{q=1}^Q \sum_{n=1}^N \Theta^q(\mu) u_{N_n}^{\mathcal{N}}(\mu) \left\{ \sum_{q'=1}^Q \sum_{n'=1}^N \Theta^{q'}(\mu) u_{N_{n'}}^{\mathcal{N}}(\mu) (\mathcal{L}_n^q, \mathcal{L}_{n'}^{q'})_X \right\}. \end{aligned} \quad (34)$$

Or de (27) on sait qu'on peut calculer la norme dual des résidus $\|r(\cdot; \mu)\|_{(X^{\mathcal{N}})^\vee}$ de la même façon.

9.3 Procédure de calcul « offline-online »

Dans la partie « offline » on va calculer l'ensemble des valeurs ne dépendant pas de $\mu \in \mathcal{D}$.

Ainsi on va commencer par déterminer les « pseudo »-solutions \mathcal{C} , \mathcal{L}_n^q , $1 \leq n \leq N_{max}$, $1 \leq q \leq Q$. Puis à partir de celles-ci on peut former et stocker les quantités suivantes :

$$(\mathcal{C}, \mathcal{C})_X, (\mathcal{C}, \mathcal{L}_n^q)_X, (\mathcal{L}_n^q, \mathcal{L}_{n'}^{q'})_X \quad 1 \leq n, n' \leq N_{max}, 1 \leq q, q' \leq Q.$$

Ainsi le nombre d'opérations « offline » dépend de N_{max} , Q et \mathcal{N} .

Dans la partie « online », pour chaque nouvelle valeur de μ , on doit évaluer $\Theta^q(\mu)$, $1 \leq q \leq Q$, $u_{N_n}^{\mathcal{N}}(\mu)$, $1 \leq n \leq N$. Puis en récupérant les $(\mathcal{C}, \mathcal{C})_X$, $(\mathcal{C}, \mathcal{L}_n^q)_X$, $(\mathcal{L}_n^q, \mathcal{L}_{n'}^{q'})_X$, $1 \leq n, n' \leq N_{max}$, $1 \leq q, q' \leq Q$ que nous avons stockés auparavant on peut évaluer l'expression (34) qui correspond à $s_N^{\mathcal{N}}(\mu)$.

Ainsi le coût de la partie « online » est de $O(Q^2 N^2)$ et donc indépendant de \mathcal{N} .

10 Mise en pratique de la méthode des bases réduites

Afin de mettre en pratique la théorie des bases réduites, nous allons nous intéresser à l'écoulement d'un fluide incompressible autour de corps solides.

Pour rendre ceci plus concret, nous allons modéliser une quille de voilier. Nous allons proposer deux modèles différents. Le premier présente une modélisation de la quille sans le support la reliant au voilier. Quand au second, il rajoute le-dit support.

Ensuite nous modéliserons un ensemble de corps pouvant représenter le fuselage d'un avion ou encore les différentes coques d'un trimaran.

Pour chacun de ces modèles, nous donnerons la modélisation du domaine. Puis nous discuterons des résultats obtenus en utilisant la méthode des bases réduites.

10.1 Le logiciel rbMIT

Afin d'effectuer les aspects calculatoire, nous nous référerons aux résultats obtenus à l'aide du programme `Matlab rbMIT` fournit sur la page

<http://augustine.mit.edu/>.

L'ensemble des représentations ont été obtenues à l'aide de ce dernier. Ce programme est par ailleurs un outil pédagogique intéressant car son installation et son utilisation sont extrêmement simples. De plus il offre une grande liberté en ce qui concerne la modélisation de problèmes. Cependant lorsqu'une erreur se produit (suite par exemple à une mauvaise modélisation du problème), le message d'erreur fournit par `Matlab` n'est pas toujours très explicite. De plus l'usage d'algorithmes génériques comme c'est le cas pour ce programme ne permet pas toujours d'obtenir les meilleures maillages possibles.

10.2 Outils de visualisation de résultats

Dans le cadre de ce projet, j'ai développé une fonction, appelé `Vis_Conv`, permettant de visualiser différentes informations concernant les différents problèmes.

Pour l'utiliser, il suffit de copier le fichier `Vis_Conv.m` dans le répertoire `rbMIT_library` du programme `rbMIT`. Le code de ce fichier est donné dans la source 5. Pour l'utiliser, il suffit de taper

```
>> Vis_Conv('rbU_probname')
```

où `probname` est le nom du problème.

10.3 Modélisation d'un domaine sous rbMIT

Le programme `rbMIT` propose aux utilisateurs deux façons de modéliser leur domaine :

- des lignes,
- des arcs elliptiques.

Il est donc essentiel de maîtriser la paramétrisation des deux types de lignes. Le premier ne posant pas de problème, passons donc au second type.

Considérons la manière suivante de définir un arc élliptique.

$$\begin{pmatrix} x_1 \\ x_2 \end{pmatrix} = \begin{bmatrix} O_1 \\ O_2 \end{bmatrix} + \begin{bmatrix} \cos \theta & -\sin \theta \\ \sin \theta & \cos \theta \end{bmatrix} \begin{bmatrix} \rho_1 & 0 \\ 0 & \rho_2 \end{bmatrix} \begin{bmatrix} \cos t \\ \sin t \end{bmatrix}$$

Cette construction est pour le moins intuitive. Le terme $(\cos t, \sin t)^T$ représente la paramétrisation du cercle unité. Ensuite on y applique une matrice de déformation (ρ_1 et ρ_2 donnent respectivement la déformation horizontale et verticale). Le résultat obtenu est une ellipse. La matrice de rotation

$$\begin{bmatrix} \cos \theta & -\sin \theta \\ \sin \theta & \cos \theta \end{bmatrix}$$

permet ensuite d'orienter l'ellipse. Enfin le vecteur $(O_1, O_2)^T$ permet de déplacer l'ellipse à l'endroit souhaité.

10.4 Un premier modèle

10.4.1 Description du modèle

Nous allons nous intéresser dans un premier temps à l'écoulement autour de la quille en la plongeant dans un fluide. Nous ometterons donc pour l'instant le support la reliant à la coque.

Pour représenter cette quille, nous allons utiliser deux demi-ellipses. Ainsi pour les besoins de notre modèle (voir Figure 1), nous introduisons les paramètres suivants :

- μ_1 Donne la hauteur de la quille,
- μ_2, μ_3 Donnent la longueur de la quille.

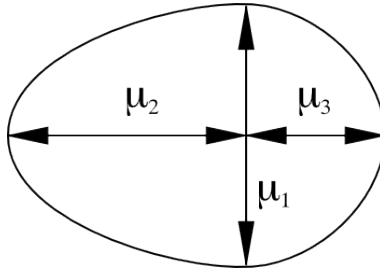


FIGURE 1 – Modélisation de l'ellipse.

10.4.2 Paramétrisation du domaine

On commence par définir une « boîte » à partir des points $(10, 10)$, $(10, 0)$, $(-10, 0)$ et $(-10, 10)$ pour contenir notre système.

Nous utilisons ensuite la modélisation vue dans la section 10.3 afin de construire la quille à partir de deux morceaux d'ellipses :

$$\begin{pmatrix} x_1 \\ x_2 \end{pmatrix} = \begin{bmatrix} 0 \\ 5 \end{bmatrix} + \begin{bmatrix} \cos \theta & -\sin \theta \\ \sin \theta & \cos \theta \end{bmatrix} \begin{bmatrix} \mu_3 & 0 \\ 0 & \frac{\mu_1}{2} \end{bmatrix} \begin{bmatrix} \cos t \\ \sin t \end{bmatrix} \quad -\frac{\pi}{2} \leq \theta < \frac{\pi}{2},$$

$$\begin{pmatrix} x_1 \\ x_2 \end{pmatrix} = \begin{bmatrix} 0 \\ 5 \end{bmatrix} + \begin{bmatrix} \cos \theta & -\sin \theta \\ \sin \theta & \cos \theta \end{bmatrix} \begin{bmatrix} \mu_2 & 0 \\ 0 & \frac{\mu_1}{2} \end{bmatrix} \begin{bmatrix} \cos t \\ \sin t \end{bmatrix} \quad \frac{\pi}{2} \leq \theta < \frac{3\pi}{2}.$$

Un aperçu du domaine est donné dans la Figure 2.

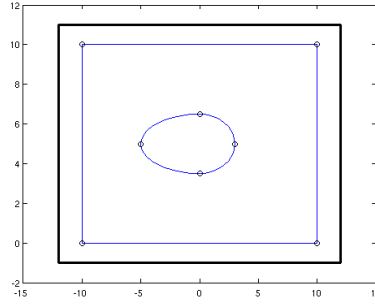


FIGURE 2 – Domaine construit par le programme.

10.4.3 Description du problème physique

Nous souhaitons simuler l'écoulement d'un fluide autour de notre quille. Pour cela, on considère le domaine Ω suivant :

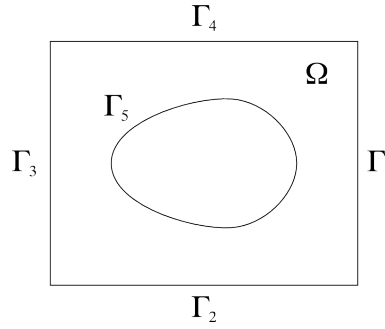


FIGURE 3 – Domaine du problème.

Nous allons supposer que notre fluide s'écoule de droite à gauche. Pour représenter ceci on impose une condition de Dirichlet sur Γ_1 (on définit la vitesse d'entrée) et une condition de Neumann sur Γ_3 (pour spécifier le flux de liquide qui s'échappe du système). Ensuite, on impose une condition de Neumann nulle sur Γ_2 , Γ_4 et Γ_5 (ceci empêche le fluide d'entrer dans la quille ou de s'échapper par les parois latérales). Ainsi notre problème, sous sa forme forte, peut s'écrire :

$$\left\{ \begin{array}{ll} \text{Trouver } u^{\mathcal{N}}(\mu) \in X^{\mathcal{N}} \text{ tel que} & \\ \Delta u^{\mathcal{N}} = 0 & \text{sur } \Omega \\ u^{\mathcal{N}} = 3 & \text{sur } \Gamma_1 \\ \partial_n u^{\mathcal{N}} = 3 & \text{sur } \Gamma_3 \\ \partial_n u^{\mathcal{N}} = 0 & \text{sur } \Gamma_2 \cup \Gamma_4 \cup \Gamma_5 \end{array} \right.$$

où ∂_n désigne la dérivé dans la direction de la normale.

10.4.4 Application de la méthode de bases réduites

Pour ce modèle et les prochains, nous allons commencer par décrire certains graphes concernant la partie « offline » puis certains concernant la partie « online ». La source 1 fournit le code d'initialisation de ce modèle.

Le programme `rbMIT` commence par construire une première triangulation (voir Figure 4(a)) sur laquelle vont se baser les calculs. Bien que notre domaine comporte un axe de symétrie, la triangulation choisie par le programme ne l'est pas. Nous reviendrons sur ce point dans la section 10.4.5. Le programme a ensuite subdivisé cette triangulation afin d'obtenir une triangulation « acceptable » (voir Figure 4(b)).

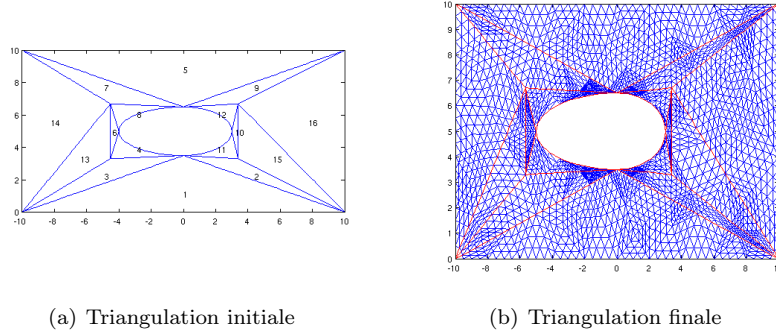


FIGURE 4 – Triangulation du domaine.

Commençons alors les procédures « offline » décrites auparavant. Nous allons considérer le graphe indiquant l'évolution de α_{LB} et α_{UB} . L'algorithme glouton pour la SCM pour la détermination de α_{LB} a nécessité 197 itérations. De plus dans la Figure 5(a), on peut constater que $\alpha_{LB} > 0$. La Figure 5(b) nous indique la réduction de l'erreur Δ_N^{en} en fonction de N (en échelle log-log) et donc par la proposition 9.1.2 la convergence de l'output. On remarque qu'ici la convergence de l'erreur est environ de l'ordre de $O(\frac{1}{n^2})$.

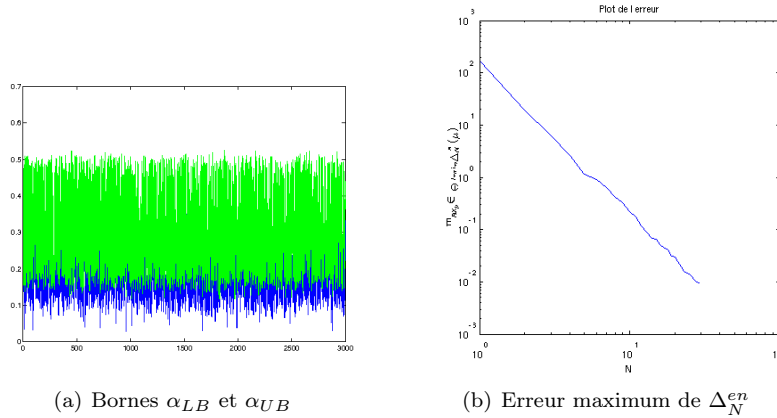


FIGURE 5 – Graphes de convergence.

En évaluant le problème pour une valeur particulière (partie « online »). Le programme nous renvoie différents outputs dont celui du potentiel, de la pression istrope et celui des (Figure 6(a), 6(b) et 6(c)).

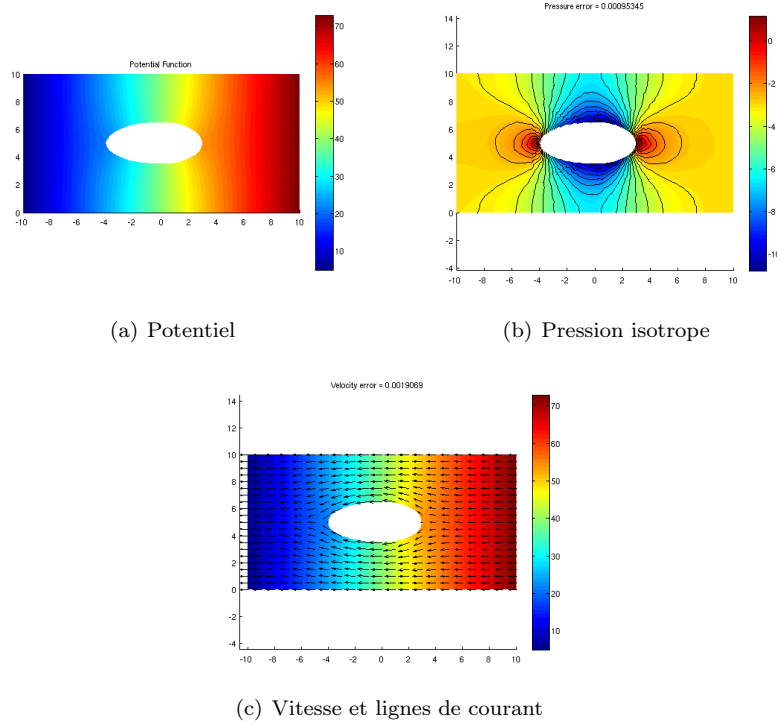


FIGURE 6 – Résultats de la simulation.

L'un des avantages de la méthode des bases réduites est de pouvoir obtenir des évaluations d'un certain paramètre $\mu \in \mathcal{D}$ rapidement. Le temps des parties « online » et « offline » est présenté dans le tableau de la Figure 10.4.4. Ainsi on peut voir que la pratique rejoint la théorie. En effet, l'effort fourni pour la partie « offline » nous permet d'obtenir extrêmement facilement l'évaluation pour μ .

	Temps de calcul (secondes)
offline	553.8
online	5.2
ratio	0.94%

FIGURE 7 – Temps de calcul

10.4.5 Optimisation du modèle

Si l'utilisation du programme `rbMIT` est relativement simple, elle nécessite une certaine vigilance de la part de l'utilisateur. Afin de permettre à l'utilisateur une certaine flexibilité dans la conception des domaines, les algorithmes utilisés se doivent d'être suffisamment souples pour les traiter. Malheureusement les algorithmes génériques ne créent pas forcément les meilleures maillages. Parfois l'ajout de points supplémentaires lors de la conception du domaine peut permettre d'améliorer le temps de calcul de la partie « offline ».

Pour illustrer ceci, reprenons le domaine que nous avons défini à la section 10.4.2. Ajoutons un point au milieu de chaque côté de la « boîte » qui contient la quille (le résultat est montré dans la Figure 8(a)). Du fait de cet ajout, le programme a cette fois construit une triangulation symétrique comme le montre la Figure 8(b).

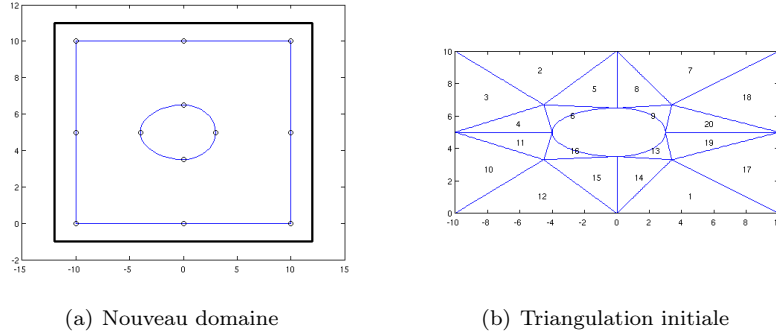


FIGURE 8 – Triangulation du domaine.

La meilleure qualité de cette triangulation initiale permet alors d'obtenir un meilleur maillage (voir Figure 10.4.5) et une meilleure convergence de l'erreur (voir Figure 9(b)). En effet le nombre d'itérations pour l'algorithme glouton pour la détermination de α_{LB} passe subitement à 30 contre 197 auparavant !

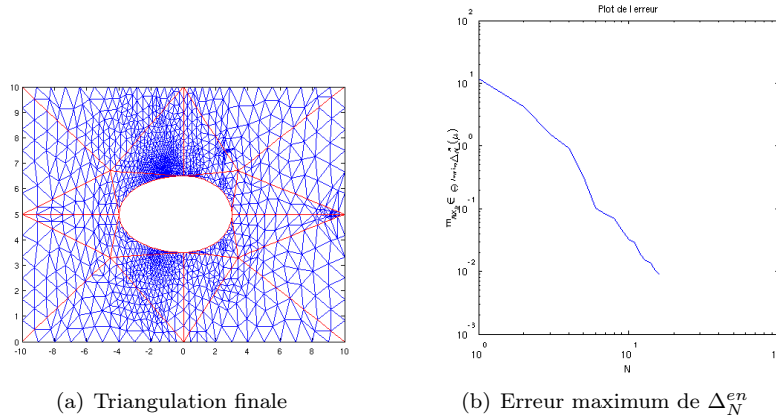


FIGURE 9 – Visualisation des améliorations.

Ceci se fait également sentir dans le temps de calcul de la partie « offline » qui passe à 395.8 secondes (environ 30% de moins du temps initial). Par contre une hausse du temps de calcul de la partie « online » est à noter (2.4 secondes en plus). L'ensemble des temps de calcul pour ce modèle sont disponibles dans la Figure 10.4.5.

	Temps de calcul (secondes)
offline	395.8
online	7.6
ratio	1.9%

FIGURE 10 – Temps de calcul

La source 2 fournit le code d'initialisation de ce modèle.

10.5 Un modèle plus élaboré

10.5.1 Description du modèle

Nous allons à présent nous intéresser à la modélisation d'une quille en ajoutant le support la reliant à la coque du bateau.

Pour notre modèle, nous allons considérer le paramètre $\mu = (\mu_1, \dots, \mu_5)^T$. La signification des paramètres est la suivante :

- μ_1 Donne la hauteur de la quille,
- μ_2, μ_3 Donnent la longueur de la quille,
- μ_4, μ_5 paramétrisent le support de la quille.

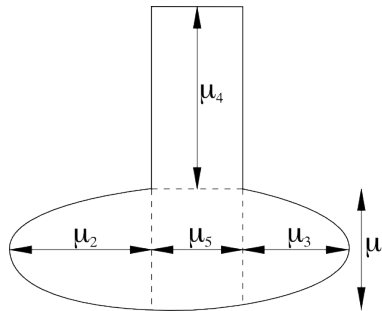


FIGURE 11 – Modélisation de la quille.

10.5.2 Paramétrisation du domaine

Comme pour le premier modèle, on définit une « boîte » à partir des points $(10, 10)$, $(10, 0)$, $(-10, 0)$ et $(-10, 10)$.

Il nous faut à présent ajouter le support et la quille au domaine. Le support ne pose aucun problème de paramétrisation : Il s'agit de deux segments de droite verticaux partant du bord dont la différence des abscisses vaut μ_5 et dont la longueur vaut μ_4 . Ainsi les points aux extrémités de ces deux segments sont

$$\left(-\frac{\mu_5}{2}, 10 - \mu_4\right) \text{ et } \left(\frac{\mu_5}{2}, 10 - \mu_4\right).$$

Ils vont nous servir à relier la quille au support. Finalement, la modélisation de

la quille va nécessiter les quatres morceaux d'ellipses suivants :

$$\begin{aligned} \begin{pmatrix} x_1 \\ x_2 \end{pmatrix} &= \begin{bmatrix} -\frac{\mu_t}{2} \\ 10 - \mu_4 - \frac{\mu_1}{2} \end{bmatrix} + \begin{bmatrix} \cos \theta & -\sin \theta \\ \sin \theta & \cos \theta \end{bmatrix} \begin{bmatrix} \mu_2 & 0 \\ 0 & \frac{\mu_1}{2} \end{bmatrix} \begin{bmatrix} \cos t \\ \sin t \end{bmatrix} & \frac{\pi}{2} \leq \theta < \pi, \\ \begin{pmatrix} x_1 \\ x_2 \end{pmatrix} &= \begin{bmatrix} 0 \\ 10 - \mu_4 - \frac{\mu_1}{2} \end{bmatrix} + \begin{bmatrix} \cos \theta & -\sin \theta \\ \sin \theta & \cos \theta \end{bmatrix} \begin{bmatrix} \mu_2 + \frac{\mu_5}{2} & 0 \\ 0 & \frac{\mu_1}{2} \end{bmatrix} \begin{bmatrix} \cos t \\ \sin t \end{bmatrix} & \pi \leq \theta < \frac{3\pi}{2}, \\ \begin{pmatrix} x_1 \\ x_2 \end{pmatrix} &= \begin{bmatrix} 0 \\ 10 - \mu_4 - \frac{\mu_1}{2} \end{bmatrix} + \begin{bmatrix} \cos \theta & -\sin \theta \\ \sin \theta & \cos \theta \end{bmatrix} \begin{bmatrix} \mu_3 + \frac{\mu_5}{2} & 0 \\ 0 & \frac{\mu_1}{2} \end{bmatrix} \begin{bmatrix} \cos t \\ \sin t \end{bmatrix} & \frac{3\pi}{2} \leq \theta < 2\pi, \\ \begin{pmatrix} x_1 \\ x_2 \end{pmatrix} &= \begin{bmatrix} \frac{\mu_t}{2} \\ 10 - \mu_4 - \frac{\mu_1}{2} \end{bmatrix} + \begin{bmatrix} \cos \theta & -\sin \theta \\ \sin \theta & \cos \theta \end{bmatrix} \begin{bmatrix} \mu_3 & 0 \\ 0 & \frac{\mu_1}{2} \end{bmatrix} \begin{bmatrix} \cos t \\ \sin t \end{bmatrix} & 0 \leq \theta < \frac{\pi}{2}. \end{aligned}$$

La figure 12 nous donne un aperçu du problème.

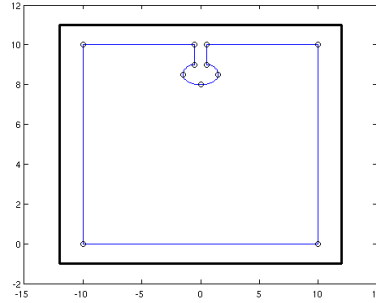


FIGURE 12 – Domaine construit par le programme.

10.5.3 Description du problème physique

La description physique du problème est exactement la même que celle explicitée dans la section 10.4.3, à l'exception du bord Γ_5 qui n'a plus la même forme. Pour ce problème, nous considérons le domaine Ω suivant :

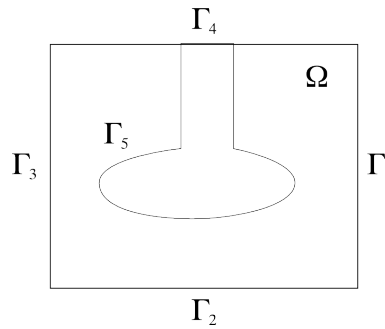


FIGURE 13 – Domaine du problème.

10.5.4 Application de la méthode de bases réduites

Commençons par décrire la triangulation du domaine (donnée Figure 14(a) et 14(b)). On remarque que l'ajout du support au domaine génère un maillage non

symétrique et surtout plus dense aux alentours de la quille et de son support. On s'aperçoit également que les triangles 6 et 16 de la Figure 14(b) sont relativement allongés et que cela va s'empirer avec la déformation du triangle si l'on étire le support d'avantage. Il serait intéressant d'ajouter des points comme nous l'avons fait pour le précédent modèle.

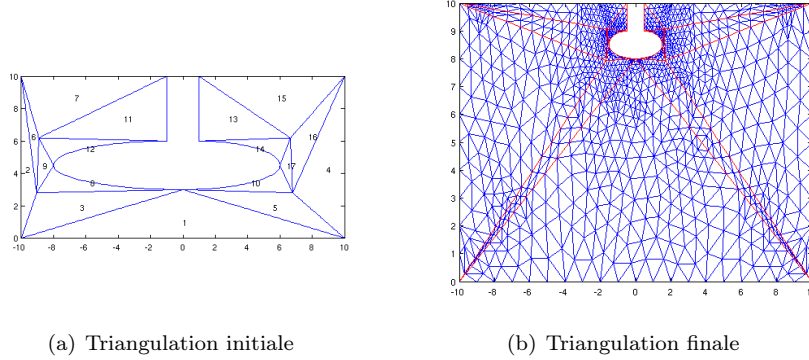


FIGURE 14 – Triangulation du domaine.

Pour ce modèle, il aura fallu 200 itérations à l'algorithme glouton pour la SCM pour la détermination de α_{LB} . Les graphes 15(a) et 15(b) donnent respectivement l'évolution des bornes α_{LB} et α_{UB} et la convergence de l'erreur.

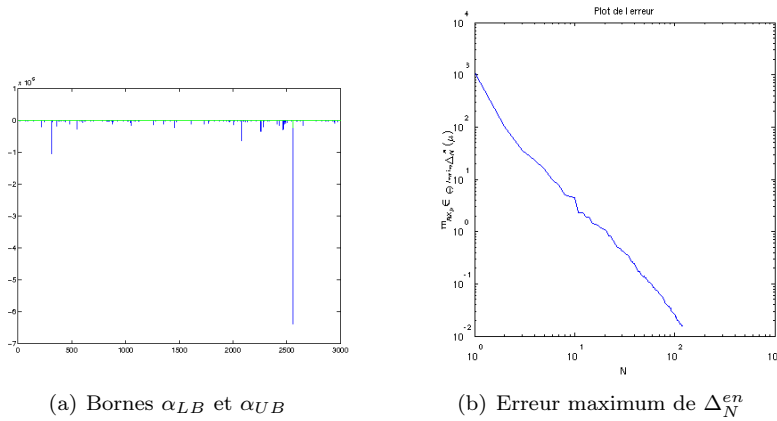


FIGURE 15 – Graphes de convergence.

La Figure 16 donne quelques outputs pour le problème de la quille. Comme on peut le voir sur la Figure 16(c), le support de la quille empêche l'écoulement du fluide au-dessus de la quille. Ainsi même si l'on change de paramètre $\mu \in \mathcal{D}$, on se rend compte qu'il existe une zone supérieure où peu de changements surviennent alors que la zone située sous la quille est soumise à différentes variations.

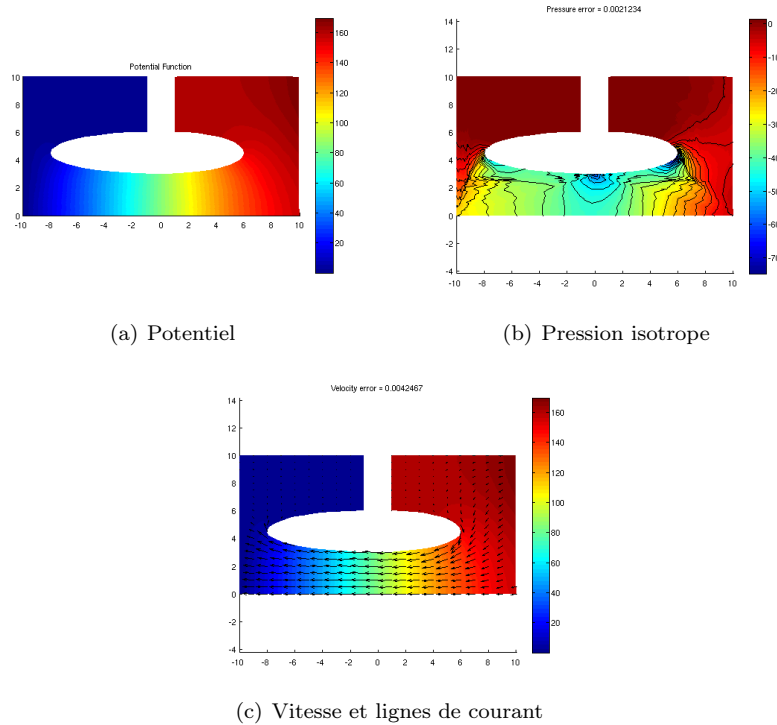


FIGURE 16 – Résultats de la simulation.

Comme auparavant, on va s'intéresser aux temps de calcul des différentes parties (voir Figure 10.5.4). Commençons par remarquer que ce modèle est plus complexe et qu'en conséquence il est plus gourmand en temps de calcul pour la partie « offline ». En effet, celui-ci est environ 20 fois plus long que la version optimisée du premier modèle.

La partie « online » nécessite par contre un temps similaire au problème optimisé. On remarque ici que le rapport entre les temps de calcul « online » et « offline » est d'environ 0.1%! Cet exemple illustre encore d'avantage les performances que peuvent atteindre la méthode des bases réduites une fois la partie « offline » terminée.

	Temps de calcul (secondes)
offline	7634.1
online	7.4
ratio	0.097%

FIGURE 17 – Temps de calcul

La source 3 fournit le code d'initialisation de ce modèle.

10.6 Ecoulement autour d'un fuselage

10.6.1 Champs d'application du modèle

Nous allons à présent nous pencher sur un modèle contenant trois corps. Ce type de configuration peut nous permettre de modéliser par exemple l'écoulement de l'aire autour de l'habitacle et des réacteurs d'un avion (Figure 18(a)) ou encore l'écoulement d'eau autour des coques d'un trimaran (Figure 18(b)).

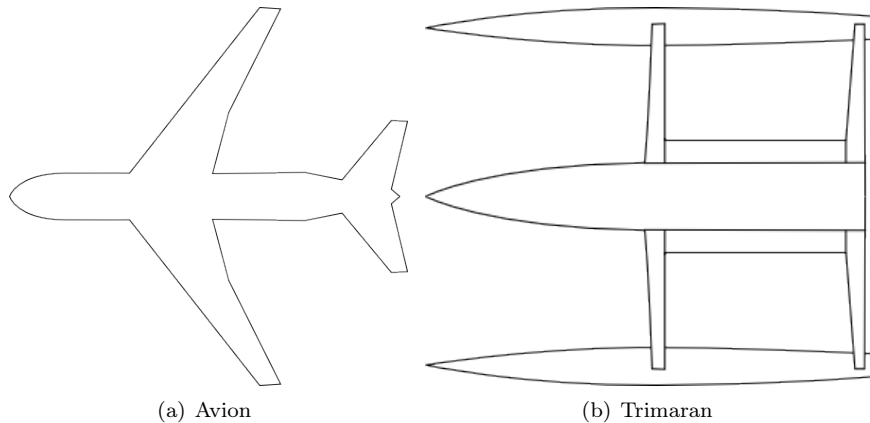


FIGURE 18 – Applications pour le modèle.

10.6.2 Description du modèle

Pour reprendre la terminologie utilisée en aéronautique, nous considérons les corps comme étant les fuselages d'un avion. Afin de les représenter mathématiquement, nous reprenons ici la modélisation à l'aide d'ellipses que nous avons utilisée précédemment. Ainsi les paramètres sont les suivants :

- μ_1, μ_2, μ_3 Donnent la hauteur et les longueurs de l'ellipse centrale,
- μ_4, μ_5, μ_6 donnent la hauteur et les longueurs des deux autres ellipses,
- μ_7 donne l'espacement vertical entre les ellipses,
- μ_8 donne le décalage horizontal de l'ellipse centrale par rapport aux deux autres.

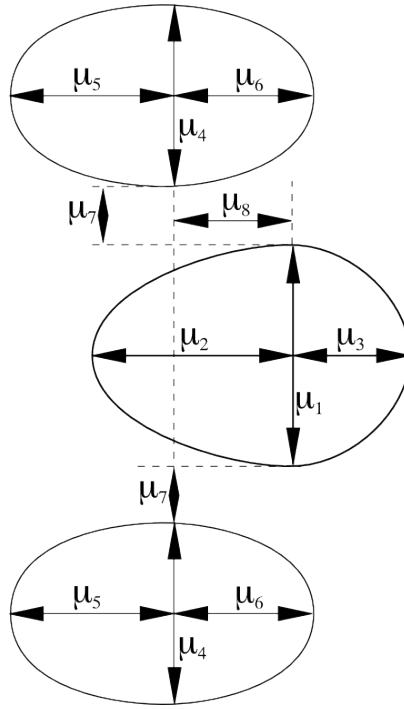


FIGURE 19 – Modélisation des fuselages.

Ce type de configurations s'avère plus difficile à traiter par les bases réduites du fait de la présence des trois fuselage (ellipses). Il est en effet important de faire attention aux différentes valeurs que peut prendre le paramètre, sinon des problèmes de chevauchement peuvent se produire. Ceci montre une fois encore que l'utilisateur doit être prudent en utilisant `rbMIT`.

10.6.3 Paramétrisation du domaine

On commence par définir une « boîte » pour contenir notre système à partir des points

$$\begin{aligned} & (30 + \mu_3 + \mu_6, 10 + \frac{\mu_1}{2} + \mu_7 + \mu_4), \\ & (30 + \mu_3 + \mu_6, -10 - \frac{\mu_1}{2} - \mu_7 - \mu_4), \\ & (-30 - \mu_2 - \mu_5, -10 - \frac{\mu_1}{2} - \mu_7 - \mu_4), \\ & (-30 - \mu_2 - \mu_5, 10 + \frac{\mu_1}{2} + \mu_7 + \mu_4). \end{aligned}$$

Nous réutilisons à nouveau la modélisation de la section 10.3 afin de construire les trois fuselages dont chacun est formé à partir de deux morceaux d'ellipses.

1. Ellipse supérieure :

$$\begin{aligned} \begin{pmatrix} x_1 \\ x_2 \end{pmatrix} &= \begin{bmatrix} 0 \\ \frac{\mu_1 + \mu_4}{2} + \mu_7 \end{bmatrix} + \begin{bmatrix} \cos \theta & -\sin \theta \\ \sin \theta & \cos \theta \end{bmatrix} \begin{bmatrix} \mu_6 & 0 \\ 0 & \frac{\mu_4}{2} \end{bmatrix} \begin{bmatrix} \cos t \\ \sin t \end{bmatrix} & -\frac{\pi}{2} \leq \theta < \frac{\pi}{2}, \\ \begin{pmatrix} x_1 \\ x_2 \end{pmatrix} &= \begin{bmatrix} 0 \\ \frac{\mu_1 + \mu_4}{2} + \mu_7 \end{bmatrix} + \begin{bmatrix} \cos \theta & -\sin \theta \\ \sin \theta & \cos \theta \end{bmatrix} \begin{bmatrix} \mu_5 & 0 \\ 0 & \frac{\mu_4}{2} \end{bmatrix} \begin{bmatrix} \cos t \\ \sin t \end{bmatrix} & \frac{\pi}{2} \leq \theta < \frac{3\pi}{2}. \end{aligned}$$

2. Ellipse centrale :

$$\begin{pmatrix} x_1 \\ x_2 \end{pmatrix} = \begin{bmatrix} \mu_8 \\ 0 \end{bmatrix} + \begin{bmatrix} \cos \theta & -\sin \theta \\ \sin \theta & \cos \theta \end{bmatrix} \begin{bmatrix} \mu_3 & 0 \\ 0 & \frac{\mu_1}{2} \end{bmatrix} \begin{bmatrix} \cos t \\ \sin t \end{bmatrix} \quad -\frac{\pi}{2} \leq \theta < \frac{\pi}{2},$$

$$\begin{pmatrix} x_1 \\ x_2 \end{pmatrix} = \begin{bmatrix} \mu_8 \\ 0 \end{bmatrix} + \begin{bmatrix} \cos \theta & -\sin \theta \\ \sin \theta & \cos \theta \end{bmatrix} \begin{bmatrix} \mu_2 & 0 \\ 0 & \frac{\mu_1}{2} \end{bmatrix} \begin{bmatrix} \cos t \\ \sin t \end{bmatrix} \quad \frac{\pi}{2} \leq \theta < \frac{3\pi}{2}.$$

3. Ellipse inférieure :

$$\begin{pmatrix} x_1 \\ x_2 \end{pmatrix} = \begin{bmatrix} 0 \\ -\frac{\mu_1 + \mu_4}{2} - \mu_7 \end{bmatrix} + \begin{bmatrix} \cos \theta & -\sin \theta \\ \sin \theta & \cos \theta \end{bmatrix} \begin{bmatrix} \mu_6 & 0 \\ 0 & \frac{\mu_4}{2} \end{bmatrix} \begin{bmatrix} \cos t \\ \sin t \end{bmatrix} \quad -\frac{\pi}{2} \leq \theta < \frac{\pi}{2},$$

$$\begin{pmatrix} x_1 \\ x_2 \end{pmatrix} = \begin{bmatrix} 0 \\ -\frac{\mu_1 + \mu_4}{2} - \mu_7 \end{bmatrix} + \begin{bmatrix} \cos \theta & -\sin \theta \\ \sin \theta & \cos \theta \end{bmatrix} \begin{bmatrix} \mu_5 & 0 \\ 0 & \frac{\mu_4}{2} \end{bmatrix} \begin{bmatrix} \cos t \\ \sin t \end{bmatrix} \quad \frac{\pi}{2} \leq \theta < \frac{3\pi}{2}.$$

Comme on l'a vu aux sections 10.4.5 et ??, l'ajout de point à des endroits judicieux peut permettre d'améliorer la convergence et les temps de calcul. Ainsi fort de cette expérience, nous allons ajouter un point au centre des bords du haut et du bas, ainsi que trois points sur les bords gauche et droit au niveau des différents fuselages. La figure 20 nous donne un aperçu du domaine obtenu.

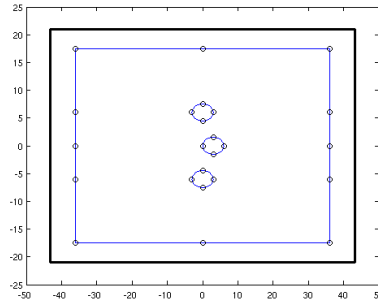


FIGURE 20 – Domaine construit par le programme.

10.6.4 Description du problème physique

Ce modèle ressemble au premier modèle que nous avons défini à la section 10.4, sauf qu'il faut considérer deux ellipses supplémentaires. Ainsi il suffit d'y ajouter deux bords Γ_6 et Γ_7 (avec condition de Neumann nulle) pour compléter la description du problème. La Figure 21 illustre le domaine Ω avec ses différents bords.

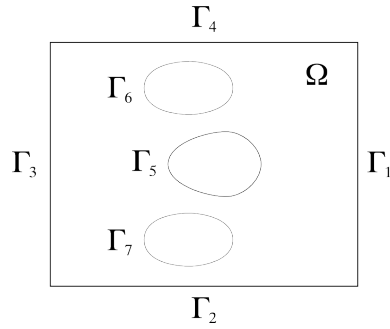


FIGURE 21 – Domaine du problème.

10.6.5 Application de la méthode des bases réduites

Ce modèle propose une interaction entre plusieurs fuselages dont la géométrie est elliptique. Ceci constitue un domaine relativement complexe engendrant une première triangulation déjà relativement concentrée autour de ces derniers (voir Figure 22(a)). Ainsi il n'est pas étonnant d'obtenir un maillage complexe et dense dans la même zone (voir Figure 22(b)).

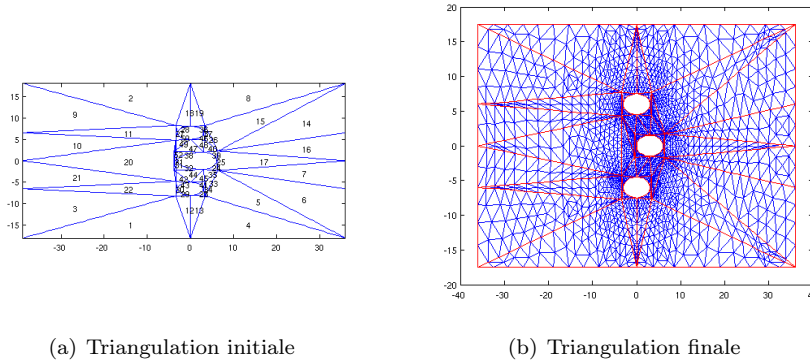


FIGURE 22 – Triangulation du domaine.

Le graphes de α_{LB} et α_{UB} et du maximum de l'erreur Δ_N^s sont présentés dans la Figure 23. La convergence l'algorithme glouton des α_{LB} a nécessité 200 itérations.

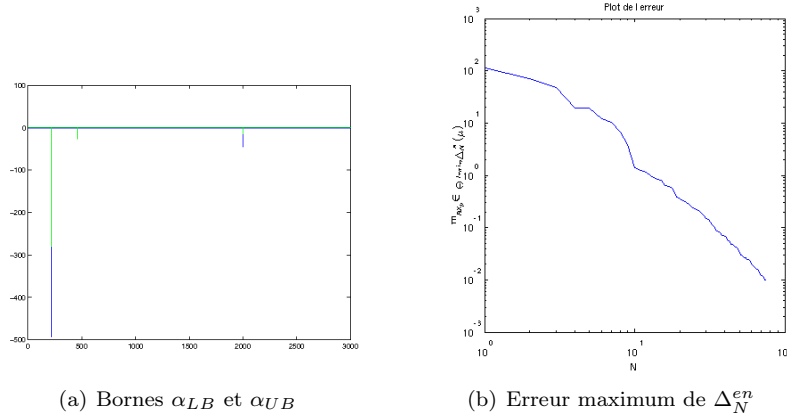


FIGURE 23 – Graphes de convergence.

La Figure 24 présente quelques outputs pour le problème. L’aspect le plus intéressant étant que le paramètre μ utilisé pour obtenir les graphes 24(b) et 24(c) faisait probablement partie des « snapshots ». On peut en effet voir sur ces graphes que l’erreur pour la pression et la vitesse est nulle.

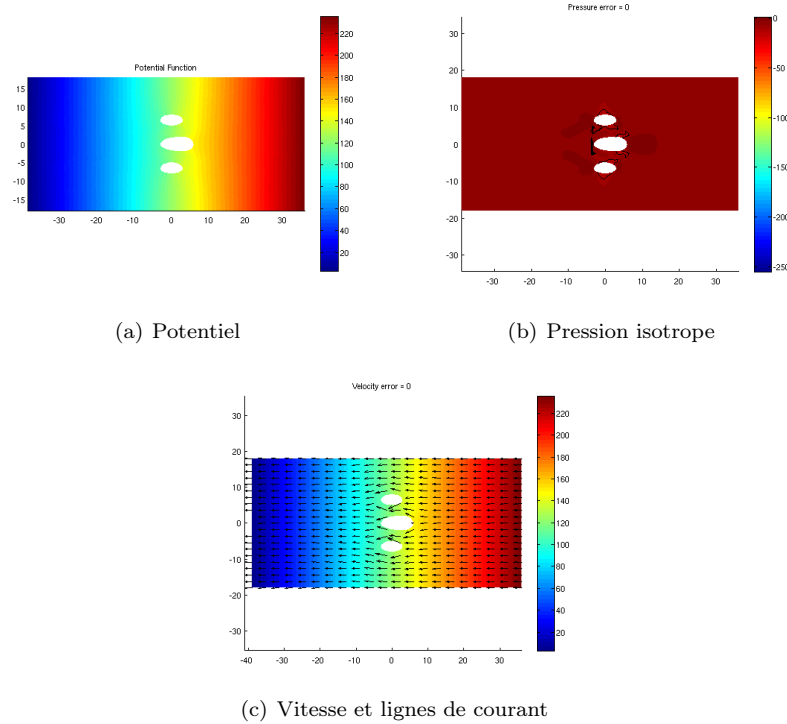


FIGURE 24 – Résultats de la simulation.

Enfin, comme pour les autres modèles, nous présentons le tableau des temps de calcul (voir Figure 10.6.5). Ce dernier illustre à nouveau le contraste entre

les parties « offline » et « online ».

	Temps de calcul (secondes)
offline	4853.7
online	16.6
ratio	0.34%

FIGURE 25 – Temps de calcul

La source 4 fournit le code d'initialisation de ce modèle.

10.7 Modèles et efforts de calcul

On souhaite à présent discuter birèvement l'ensemble des temps obtenus pour les trois modèles. Ces temps sont donnés dans la Figure 10.7.

	Temps de calcul (secondes)		
	Quille (opt)	Quille + support	Fuselages
offline	395.8	7634.1	4853.7
online	7.6	7.4	16.6
ratio	1.9%	0.097%	0.34%

FIGURE 26 – Temps de calcul

Nous avons introduit un premier modèle qui consistait en deux ellipses. Ce simple modèle va nous servir de point de référence pour les autres modèles. La méthode des bases réduites fait intervenir de nombreux sujets d'analyse numérique dont les maillages et la méthode des éléments finis. Ce sont autant de contraintes qui font que plus un modèle devient complexe, plus les coûts deviennent élevés. Bien que ceci ne constitue pas une preuve, ce phénomène se manifeste dans les modèles proposés.

En effet le modèle des fuselages peut être vu comme une extension du premier modèle dans lequel on aurait créé deux nouvelles formes composées de morceaux d'ellipses. Dans la pratique on se rend bien compte que le temps de calcul est largement supérieur au temps qu'il aurait fallu pour résoudre séparément les problèmes liés aux trois fuselages. Cette augmentation vient de l'interaction entre les différents objets (aussi bien au niveau physique qu'au niveau des contraintes liées au maillage).

De même le « simple » ajout du support à la quille a suffi à rendre le modèle complexe. Cette partie du domaine a en effet une grande influence sur l'écoulement : elle empêche le fluide de passer au-dessus de la quille. C'est d'ailleurs le modèle le plus coûteux que nous ayons considéré.

Du point de vue de la méthode des bases réduites, cette explosion des temps vient du fait que pour obtenir une tolérance satisfaisante pour l'algorithme glouton, nous devons posséder de plus en plus de « snapshots ». Ainsi plus le problème est complexe et réagit à la variation des paramètres plus il faudra en calculer.

11 Conclusion

Durant ce projet nous avons développé les concepts fondamentaux concernant la méthode des bases réduites. Nous avons commencé par introduire un problème modèle pour lequel nous avons développé une stratégie permettant de diviser les coûts de calcul en deux : Les parties « offline » et « online ».

Nous nous sommes ensuite penché sur la théorie nécessaire pour fournir des estimateurs pour l'erreur a posteriori. Nous avons également introduit la notion d'efficacité pour donner une mesure de la qualité de ces estimateurs.

Finalement nous avons présenté le logiciel `rbMIT` avec lequel nous avons étudié trois exemples mettant en oeuvre la méthode des bases réduites.

Pour conclure, nous insistons sur le fait que ce document n'est qu'une introduction au sujet et que la méthode peut être étendue à des problèmes plus complexes tels que ceux d'advection-diffusion. Il est également possible de considérer des domaines évoluant au fil du temps.

A Fichiers sources pour les modèles proposés

```

clear all;
% License material
%
% -----rbMIT Software Copyright MIT 2006-07-----
% -----DBP Huynh, NC Nguyen, AT Patera, G Rozza-----
%
% The user agrees to rbMIT ??? MIT usage, distribution, and citation
% terms and conditions
% described in the License at http://augustine.mit.edu/
% rbMIT_SystemLicense.htm.
% rbMIT Software Part_II_and_IV
%
% This example was developped by Gweno1 Grandperrin
% 18th november 2008

plotdemo = 0;
% This problem deals with a keel in a fluid
probnome = 'keel1'

points = '[10,10;10,0;-10,0;-10,10;0,5+mu1/2;-mu2,5;0,5-mu1/2;mu3,5]';

edge = [1,2;2,3;3,4;4,1;5,6;6,7;7,8;8,5];
geometry{1} = [1,2,3,4];
geometry{2} = [5,6,7,8];
gflag = [1,0]; %geometry full/empty
kappa{1} = '[1 0 0; 0, 1, 0; 0 0 0]';
kappa{2} = '[0 0 0; 0 0 0; 0 0 0]';
muref = [3,4,3]; % valeur de ref
mu_min = [1,1,1]; % minimal value
mu_max = [4,4,4]; % maximal value
mu_bar = [2,2,2]; % ?

%condition de Neumann
nload = '[1,0,0,3]'; %edge 1->right value 3
% flux conditions on the left boundary
dirichlet = '[3,5]'; %flux value 5

% zero Dirichlet conditions on the left boundary
curvedat = '[0,5,mu3,mu1/2,0,cos(t),sin(t);0,5,mu2,mu1/2,0,cos(t),sin(t)
;0,5,mu2,mu1/2,0,cos(t),sin(t);0,5,mu3,mu1/2,0,cos(t),sin(t)]';
tarclist = '[5,2,pi/2,pi;6,3,pi,3/2*pi;7,4,3/2*pi,2*pi;8,1,0,pi/2]';

outputname = 'none'
oload = '[4,1]'; % just a random output

femOPT.refine = 'subdivision';
femOPT.maxsize = 1000;
pr_DEF.type='potential';
%%%%%%%%%%%% no user input required beyond this point
if ~exist('plotdemo')
    plotdemo = 0;

```

```

end

save rbU ;
copyfile('../rbMIT_Aux/Step1_coer_noncompliant.m',strcat(probname,'_',
Step1_coer_noncompliant.m'))
eval(strcat(probname,'_',Step1_coer_noncompliant'))

```

Source 1 – rbU_keel1.m

```

clear all;
% License material
%
% -----rbMIT Software Copyright MIT 2006-07-----
% -----DBP Huynh, NC Nguyen, AT Patera, G Rozza-----
%
% The user agrees to rbMIT ??? MIT usage, distribution, and citation
% terms and conditions
% described in the License at http://augustine.mit.edu/
% rbMIT\_SystemLicense.htm.
% rbMIT Software Part_II_and_IV
%
% This example was developed by Gwenol Grandperrin
% 18th november 2008

plotdemo = 0;
% This problem deals with a keel in a fluid
probname = 'keel1opt'

points = '[10,10;10,5;10,0;0,0;-10,0;-10,5;-10,10;0,10;0,5+mu1/2;-mu2
,5;0,5-mu1/2;mu3,5]';

edge = [1,2;2,3;3,4;4,5;5,6;6,7;7,8;8,1;9,10;10,11;11,12;12,9];
geometry{1} = [1,2,3,4,5,6,7,8];
geometry{2} = [9,10,11,12];
gflag = [1,0]; %geometry full/empty
kappa{1} = '[1 0 0; 0, 1, 0; 0 0 0]';
kappa{2} = '[0 0 0; 0 0 0; 0 0 0]';
muref = [3,4,3]; % valeur de ref
mu_min = [1,1,1]; % minimal value
mu_max = [4,4,4]; % maximal value
mu_bar = [2,2,2]; % ?

%condition de Neumann
nload = '[1,0,0,3;2,0,0,3]'; %edge 1,2->right value 3
% flux conditions on the left boundary
dirichlet = '[5,3;6,3]'; %flux value 3

% zero Dirichlet conditions on the left boundary
curvedat = '[0,5,mu3,mu1/2,0,cos(t),sin(t);0,5,mu2,mu1/2,0,cos(t),sin(t)]
';
tarclist = '[9,2,pi/2,pi;10,2,pi,3/2*pi;11,1,3/2*pi,2*pi;12,1,0,pi/2]';

```

```

outputname = 'none'
oload = '[4,1]'; % just a random output

femOPT.refine = 'subdivision';
femOPT.maxsize = 1000;
pr_DEF.type='potential';
%%%%%%%%%% no user input required beyond this point
if ~exist('plotdemo')
    plotdemo = 0;
end

save rbU ;
copyfile('./rbMIT_Aux/Step1_coer_noncompliant.m',strcat(probname,'_',
    Step1_coer_noncompliant.m'))
eval(strcat(probname,'_',Step1_coer_noncompliant'))

```

Source 2 – rbU_keel1opt.m

```

clear all;
% License material
%
% -----rbMIT Software Copyright MIT 2006-07-----
% -----DBP Huynh, NC Nguyen, AT Patera, G Rozza-----
%
% The user agrees to rbMIT ??? MIT usage, distribution, and citation
    terms and conditions
% described in the License at http://augustine.mit.edu/
    rbMIT_SystemLicense.htm.
% rbMIT Software Part_II_and_IV
%
%This example was developed by Gwenoł Grandperrin
% 18th november 2008

plotdemo = 0;
% This problem deals with a keel in a fluid
probname = 'keel2'

%111 iterations

points = '[10,10;10,0;-10,0;-10,10;-mu5/2,10;-mu5/2,10-mu4;-mu2-mu5/2,10-
    mu4-mu1/2;0,10-mu4-mu1;mu3+mu5/2,10-mu4-mu1/2;mu5/2,10-mu4;mu5/2,10]
    ';

edge = [1,2;2,3;3,4;4,5;5,6;6,7;7,8;8,9;9,10;10,11;11,1];
geometry{1} = [1,2,3,4,5,6,7,8,9,10,11];
gflag = [1]; %geometry full/empty
kappa{1} = '[1 0 0; 0, 1, 0; 0 0 0]';
muref = [1,1,1,1,1]; % valeur de ref
%[3,7,5,4,2]
mu_min = [1,1,1,1,1]; % minimal value
mu_max = [7,7,7,8,5]; % maximal value
mu_bar = [3,5,3,4,3]; % ?

```

```

nload = '[1,0,0,3]';
% flux conditions on the left boundary
dirichlet = '[3,0]';
% zero Dirichlet conditions on the left boundary
curvedat = '[mu5/2,10-mu4-mu1/2,mu3,mu1/2,0,cos(t),sin(t);-mu5/2,10-mu4-
mu1/2,mu2,mu1/2,0,cos(t),sin(t);0,10-mu4-mu1/2,mu2+mu5/2,mu1/2,0,cos
(t),sin(t);0,10-mu4-mu1/2,mu3+mu5/2,mu1/2,0,cos(t),sin(t)]';
tarclist = '[9,1,0,pi/2;6,2,pi/2,pi;7,3,pi,3*pi/2;8,4,3*pi/2,2*pi]';

outputname = 'none'
oload = '[4,1]'; % just a random output

femOPT.refine = 'subdivision';
femOPT.maxsize = 1000;
pr_DEF.type='potential';
%%%%%%%%%% no user input required beyond this point
if ~exist('plotdemo')
    plotdemo = 0;
end

save rbU ;
copyfile('../rbMIT_Aux/Step1_coer_noncompliant.m',strcat(probname,'_',
    Step1_coer_noncompliant.m'))
eval(strcat(probname,'_',Step1_coer_noncompliant'))

```

Source 3 – rbU_keel2.m

```

clear all;
% License material
%
% -----rbMIT Software Copyright MIT 2006-07-----
% -----DBP Huynh, NC Nguyen, AT Patera, G Rozza-----
%
% The user agrees to rbMIT ??? MIT usage, distribution, and citation
    terms and conditions
% described in the License at http://augustine.mit.edu/
    rbMIT_SystemLicense.htm.
% rbMIT Software Part_II_and_IV
%
%This example was developped by Gwenol Grandperrin
% 18th november 2008

plotdemo = 0;
% This problem deals with 3 geometries representing a plane fuselage
probname = 'fuselage'

%4 pts for the central fuselage
%8 pts for the 2 other fuselage (4 pts each)
%12 pts for the domain boundaries
%10 minimal space between object and bordery
points = '[30+mu3+mu6,10+mu1/2+mu7+mu4;30+mu3+mu6,(mu1+mu4)/2+mu7;30+mu3+
mu6,0;30+mu3+mu6,-(mu1+mu4)/2-mu7;30+mu3+mu6,-10-mu1/2-mu7-mu4
;0,-10-mu1/2-mu7-mu4;-30-mu2-mu5,-10-mu1/2-mu7-mu4;-30-mu2-mu5,-(mu1

```

```

+mu4)/2-mu7;-30-mu2-mu5,0;-30-mu2-mu5,(mu1+mu4)/2+mu7;-30-mu2-mu5
,10+mu1/2+mu7+mu4;0,10+mu1/2+mu7+mu4;mu8+mu3,0;mu8,mu1/2;mu8-mu2,0;
mu8,-mu1/2;mu6,mu1/2+mu7+mu4/2;0,mu1/2+mu7+mu4;-mu5,mu1/2+mu7+mu4
/2;0,mu1/2+mu7;mu6,-mu1/2-mu7-mu4/2;0,-mu1/2-mu7;-mu5,-mu1/2-mu7-mu4
/2;0,-mu1/2-mu7-mu4';

```

```
edge =
```

```
[1,2;2,3;3,4;4,5;5,6;6,7;7,8;8,9;9,10;10,11;11,12;12,1;13,14;14,15;15,16;16,13;17,18;18,19;19,20;
```

```
geometry{1} = [1,2,3,4,5,6,7,8,9,10,11,12]; %Bordery
```

```
geometry{2} = [13,14,15,16]; %Central fuselage
```

```
geometry{3} = [17,18,19,20]; %Upper fuselage
```

```
geometry{4} = [21,22,23,24]; %Lower fuselage
```

```
gflag = [1,0,0,0]; %geometry full/empty
```

```
kappa{1} = '[1 0 0; 0, 1, 0; 0 0 0]';
```

```
kappa{2} = '[0 0 0; 0 0 0; 0 0 0]';
```

```
kappa{3} = '[0 0 0; 0 0 0; 0 0 0]';
```

```
kappa{4} = '[0 0 0; 0 0 0; 0 0 0]';
```

```
%works well
```

```
% Vis_RE('fuselage',[4,6,3,3,3,3,3,3]);
```

```
muref = [3,3,3,3,3,3,3,3];
```

```
mu_min = [3,3,3,3,3,3,3,3]; % minimal value
```

```
mu_max = [6,6,6,6,6,6,8,8]; % maximal value
```

```
mu_bar = [3,3,3,3,3,3,3,3]; % ?
```

```
%muref = [8,6,16,8,6,16,20,-3]; % valeur de ref
```

```
% muref = [20,20,20,20,20,20,8,8];
```

```
% mu_min = [3,3,3,3,3,3,3,-8]; % minimal value
```

```
% mu_max = [20,20,20,20,20,20,8,8]; % maximal value
```

```
% mu_bar = [3,3,3,3,3,3,3,3]; % ?
```

```
%condition de Neumann
```

```
nload = '[1,0,0,3;2,0,0,3;3,0,0,3;4,0,0,3]'; %right border value 3
```

```
% flux conditions on the left boundary
```

```
dirichlet = '[7,3;8,3;9,3;10,3]'; %flux left border value 3
```

```
%We define the curving edge for the 3 fuselages
```

```
curvedat = '[mu8,0,mu3,mu1/2,0,cos(t),sin(t);mu8,0,mu2,mu1/2,0,cos(t),sin
(t);0,(mu1+mu4)/2+mu7,mu6,mu4/2,0,cos(t),sin(t);0,(mu1+mu4)/2+mu7,
mu5,mu4/2,0,cos(t),sin(t);0,-(mu1+mu4)/2-mu7,mu6,mu4/2,0,cos(t),sin(
t);0,-(mu1+mu4)/2-mu7,mu5,mu4/2,0,cos(t),sin(t)]';
```

```
tarclist = '[13,1,0,pi/2;14,2,pi/2,pi;15,2,pi,3/2*pi;16,1,3/2*pi,2*pi
;17,3,0,pi/2;18,4,pi/2,pi;19,4,pi,3/2*pi;20,3,3*pi/2,2*pi;21,5,0,pi
/2;22,6,pi/2,pi;23,6,pi,3/2*pi;24,5,3/2*pi,2*pi]';
```

```
outputname = 'none'
```

```
oload = '[4,1]'; % just a random output
```

```
femOPT.refine = 'subdivision';
```

```
femOPT.maxsize = 1000;
```

```
pr_DEF.type='potential';
```

```
%%%%%%%%%% no user input required beyond this point
```

```
if ~exist('plotdemo')
    plotdemo = 0;
end

save rbU ;
copyfile('../rbMIT_Aux/Step1_coer_noncompliant.m',strcat(probname,'_',
    Step1_coer_noncompliant.m'))
eval(strcat(probname,'_',Step1_coer_noncompliant'))
```

Source 4 – rbU_fuselage.m

```
function Vis_Conv(prob_name)
% Plot the lower and upper bounds for the
% coercivity constant \alpha
% -----
% Author: Gwenol Grandperrin
% Date : 04.11.08
% -----
% Usage : Vis_Conv('prob_name')

%Loading data
disp('Loading data...');
file1 = [prob_name,'/',prob_name,'_pr_ONLINE.mat'];
%file2 = [prob_name,'/',prob_name,'_pr_OFFLINE.mat'];
load(file1);
%load(file2);

%Getting selected value of the parameters
%figure(100);
%Mus = eval([prob_name,'_pr_ONLINE.space.Mus']);
%plot(Mus);

%Getting the max error bound for the greedy algorithm (\Delta_N^s(\mu))
figure(101);
greedyMaxErrorBounds = eval([prob_name,'_pr_ONLINE.space.eps_out_star']);
plot(greedyMaxErrorBounds);
xlabel('N')
ylabel('$\max_{\mu \in \Theta_{\text{train}}}\Delta_N^s(\mu)$','Interpreter','latex')
title('Plot de l erreur')

%Affichage du nombre d'iterations
disp(['Nb d iterations: ',num2str(length(greedyMaxErrorBounds))]);

%Plot the coercivity bounds
figure(102);
bound(1,:) = eval([prob_name,'_pr_ONLINE.coerlb.sigmaUB']);
bound(2,:) = eval([prob_name,'_pr_ONLINE.coerlb.sigmaLB']);
%sample_in_basis = eval([prob_name,'_pr_OFFLINE.space.sample_in_basis']);
plot(bound(1,:));
hold on;
plot(bound(2:,:), '--r');
hold off;
```

```
xlabel('N')
ylabel('$\alpha_{LB}^{\mathcal{N}}$ (en bas), $\alpha_{UB}^{\mathcal{N}}$ (en
    haut), ', 'Interpreter', 'latex')
title('Plot des bornes pour la constante de coercivité')

%Plot the coercivity bounds
figure(103);
alpha = eval([prob_name, '_pr_ONLINE.coerlb.alpha_bar']);
%sample_in_basis = eval([prob_name, '_pr_OFFLINE.space.sample_in_basis']);
plot(alpha);

xlabel('N')
ylabel('$\frac{\alpha_{UB}^{\mathcal{N}} - \alpha_{LB}^{\mathcal{N}}}{\alpha_{
    UB}^{\mathcal{N}}}$', 'Interpreter', 'latex')
title('Plot du rapport entre les bornes de la constante de coercivité')

%Affichage de la taille de [-]train
%disp(['Nb d elements de $n_{train}$: ', num2str(length(sample_in_basis))
    ], 'Interpreter', 'latex');
return
```

Source 5 – Vis_Conv.m

Références

- [1] M. Discacciati. *Numerical Approximation of Partial Differential Equations (cours)*. 2008.
- [2] A. Quarteroni et A. Valli. *Numerical Approximation of Partial Differential Equations*. Number 23. Springer, 1997.
- [3] A. T. Patera G. Rozza, D. B. P. Huynh. Reduced basis approximation and a posteriori error estimation for affinely parametrized elliptic coercive partial differential equations. *Arch Comput Methods Eng manuscript*, 2007.
- [4] S. S. Ravindran K. Ito. A reduced-order method for simulation and control of fluid flows. *Journal of Computational Physics*, (143) :403–425, 1998.
- [5] J. S. Peterson. The reduced-basis method for incompressible viscous flow calculations. *SIAM J. Sci. Stat. Comput.*, (4) :777–786, 1989.
- [6] T. A. Porsching. The reduced-basis method for initial value problems. *SIAM Journal of Numerical Analysis*, (24) :1277–1287, 1987.