

Algorithme Glouton faible par apprentissage

Congo Job

Stage Master 1 de juin à juillet 2022 sous la direction de Joubine Aghili
IRMA, Université de Strasbourg, France

22 août 2022



Table des matières

1	Introduction	3
1.1	Contexte du stage	3
1.2	Objectif du stage	3
1.3	Etat de l'art	4
1.4	Plan du rapport	4
2	Elements finis	4
2.1	Formulation Variationnelle	4
2.2	Discrétisation du problème	5
2.2.1	Construction des espaces d'approximation	5
2.2.2	Construction des matrices éléments finis	6
2.3	Validation	8
3	Modèle réduit	11
3.1	Formulation du problème réduit	11
3.2	Espace des bases réduites V_{N_0}	11
3.3	Matrices réduites	12
3.4	Etude du modèle réduit	13

1 Introduction

1.1 Contexte du stage

Ce stage est réalisé à Institut de Recherche de Mathématiques Avancées (IRMA). L'IRMA, créé en 1966, est depuis 1997 une unité mixte de recherche sous la double tutelle du CNRS et de l'Université de Strasbourg. Il fait partie de l'UFR de Mathématique et Informatique. Elle compte quelques 130 membres, dont 87 chercheurs et enseignants-chercheurs permanents et une quarantaine de non-permanents répartis en 7 équipes de recherche. C'est avec l'équipe "Modélisation et contrôle" que le stage a eu lieu. Cette équipe s'intéresse à l'analyse des EDP, à l'optimisation, à la théorie du contrôle, au calcul scientifique et haute performance, et à la statistique. Ce stage a eu lieu du 1 juin 2022 au 31 juillet 2022. On retrouvera tous les documents du stage dans ce lien : <https://github.com/master-csmi/2022-stage-job-br>.

1.2 Objectif du stage

Les méthodes numériques ont fait leurs preuves depuis quelques années. Cependant, dans le domaine du calcul scientifique, 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 considère un problème dont la solution dépend de paramètres. Ainsi partant d'une valeur pour ces paramètres, elle permet d'obtenir la solution correspondante après un temps de calcul plus court.

L'objectif de ce stage est de construire un algorithme Glouton faible avec une modèle entraîné par réseau de neurone. Une première étape consiste à implémenter un exemple en 1d en Python avec la méthode des éléments finis et donner une validation, puis implémenter la méthode des bases réduites et donner une validation, enfin entraîner un réseau de neurone apprenant l'erreur de projection à l'aide de PyTorch/TensorFlow. Par la suite effectuer la même chose sur un exemple 2D.

Pour ce faire, on commencera par s'intéresser à une EDP en 1D de type réaction-diffusion s'écrivant :

$$\begin{cases} -\frac{\partial}{\partial x}(D \cdot \frac{\partial u}{\partial x}) + u = f & \Omega \\ u = 0 & \partial\Omega \\ D(x) = \begin{cases} 1, x \in \Omega_1 \\ \mu, x \in \Omega_2 \end{cases} & (1.2) \\ \text{Avec } \Omega =]0, 1[\quad \Omega_1 = \Omega \setminus \Omega_2 \\ \Omega_2 =]0.19; 0.21[\cup]0.39; 0.41[\cup]0.59; 0.61[\cup]0.79; 0.81[\end{cases}$$

On résoudra le problème 1D (1.2) par la méthode des éléments finis et en même temps on construira un solveur $U(\mu)$. Cette fonction $U(\mu)$ renverra pour un μ donné la solution élément fini associé. Puis nous mettrons en œuvre la méthode des bases réduites à l'aide de cette fonction $U(\mu)$ et de l'algorithme Glouton fort. L'algorithme Glouton fort utilise les

solutions ainsi obtenues par $U(\mu)$ pour engendrer un espace vectoriel de dimension réduite $V_{N_0} = Vect(u_1, u_2, \dots, u_{N_0})$. Le travail principal de cet algorithme consiste à maximiser l'erreur de projection $\Delta(V_{N_0}, \mu) = \|u(\mu) - P_{V_{N_0}} u(\mu)\|$. Enfin, nous entraînerons par réseau de neurone un modèle $\Delta^{NN}(\mu)$ qui remplacera $\Delta(V_{N_0}, \mu)$ dans l'algorithme Greedy Fort. Nous espérons alors, grâce à ce modèle, évaluer rapidement l'erreur pour tout point μ , et de profiter de son caractère différentiable pour obtenir une base réduite de meilleure qualité après optimisation.

1.3 Etat de l'art

1.4 Plan du rapport

Dans la section 2, nous résolvons le problème élément finis et ferons une étude de convergence, dans la section 3 la résolution du modèle réduit avec l'algorithme Glouton fort.

2 Elements finis

On s'intéresse au problème 1D suivant de [3] :

$$\begin{cases} -\frac{\partial}{\partial x}(D \cdot \frac{\partial u}{\partial x}) + u = f, & \Omega \\ u = 0, & \partial\Omega \\ D(x) = \begin{cases} 1, x \in \Omega_1 \\ \mu, x \in \Omega_2 \end{cases} \end{cases} \quad (1)$$

2.1 Formulation Variationnelle

On multiplie (1) par $v \in H_0^1(\Omega)$ et intègre sur Ω :

$$\int_{\Omega} -v(D \cdot u')' + \int_{\Omega} vu = \int_{\Omega} fv \quad (2)$$

Après intégration par partie où on a utilisé $v|_{\partial\Omega} = 0$, on a :

$$\int_{\Omega} D \cdot v' u' + \int_{\Omega} vu = \int_{\Omega} fv$$

On pose :

$$\forall u, v \in H_0^1(\Omega) \quad a(u, v; \mu) := \int_{\Omega} \mathbb{1}_{\Omega_1} v' u' + \mu \int_{\Omega} \mathbb{1}_{\Omega_2} v' u' + \int_{\Omega} vu, \quad l(v) := \int_{\Omega} fv.$$

Et la formulation variationnelle de (2) se réécrit, trouver $u \in H_0^1(\Omega)$ tel que :

$$a(u, v; \mu) = l(v), \forall v \in H_0^1(\Omega) \quad (3)$$

2.2 Discrétisation du problème

On va décrire le problème éléments finis pour le problème (3) avec une approximation \mathbb{P}^1 , pour plus d'informations voir [Grégoire Allaire, Analyse num et optimisation]

2.2.1 Construction des espaces d'approximation

Soit Ω un segment, que l'on décompose en Ω_1 et Ω_2 disjoints et construit un maillage équidistant :

$$\Omega = \bigcup_{i=0}^{Nel-1} K_i$$

tel que :

1. $\Omega_2 =]0.19, 0.21[\cup]0.39, 0.41[\cup]0.59, 0.61[\cup]0.79, 0.81[$ et $\Omega_1 = \Omega \setminus \Omega_2$.
2. Nel est le nombre d'éléments du maillage et h le pas constant vaut $\frac{1}{Nel-1}$.
3. K_i la maille $[x_i, x_{i+1}]$ avec $x_i = ih, \forall i \in \{0, \dots, Nel-1\}$.
4. On suppose que chaque maille K_i se trouve soit dans $\overline{\Omega_1}$ soit dans $\overline{\Omega_2}$.

Après la décomposition de Ω , nous construisons l'espace d'approximation V_h des fonctions $\varphi_i : \Omega \rightarrow \mathbb{R}$ de la manière suivante :

Après la décomposition de Ω , nous construisons l'espace d'approximation V_h des fonctions $\varphi_i : \Omega \rightarrow \mathbb{R}$ de la manière suivante :

$$\forall i \in [0, Nel-1], \varphi_i(x) = \begin{cases} \frac{x-x_{i-1}}{h}, & x \in [x_{i-1}, x_i] \\ -\frac{x-x_{i+1}}{h}, & x \in [x_i, x_{i+1}] \\ 0, & \text{sinon} \end{cases}$$

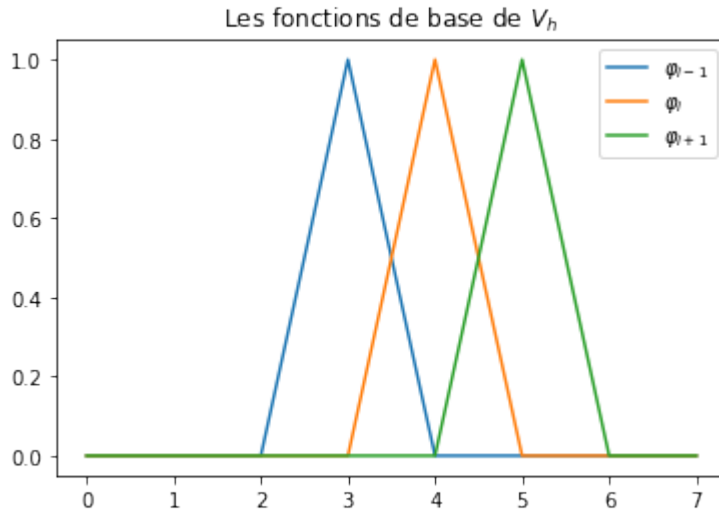


FIGURE 1 – Illustration des fonctions de base

Et sa dérivée est définie, lorsque c'est possible, par la formule :

$$\forall i \in [0, Nel - 1], \varphi'_i(x) = \begin{cases} \frac{1}{h}, x \in]x_{i-1}, x_i[\\ -\frac{1}{h}, x \in]x_i, x_{i+1}[\\ 0, \text{sinon} \end{cases}$$

L'espace d'approximation \mathbb{P}^1 est alors :

$$V_h = Vect(\{\varphi_i\})$$

Comme V_h est l'espace d'approximation \mathbb{P}^1 alors une discrétisation possible de $H_0^1(\Omega)$ est l'espace $V_{h,0}$ défini par :

$$V_{h,0} = \{u \in V_h \mid \forall i \in \{0, \dots, Nel - 1\} \ u_{|K_i} \in \mathbb{P}^1(\Omega) \text{ et } u_{|\partial\Omega} = 0\}$$

2.2.2 Construction des matrices éléments finis

En remplaçant $H_0^1(\Omega)$ par $V_{h,0}$, on va chercher à résoudre le problème approché :

$$\text{Trouver } u_h \in V_{h,0} \text{ tel que : } a(u_h, v_h; \mu) = l(v_h), \forall v_h \in V_{h,0} \quad (4)$$

On sait que $\forall u_h \in V_{h,0}$, $u_h = \sum_{i=0}^{Nel-1} \alpha_i \varphi_i$ donc (4) devient

$$\sum_{i=0}^{Nel-1} \alpha_i a(\varphi_i, v_h; \mu) = l(v_h), \forall v_h \in V_{h,0}$$

En prenant $v_h = \varphi_j$ on a :

$$\sum_{i=0}^{Nel-1} \alpha_i a(\varphi_i, \varphi_j; \mu) = l(\varphi_j), \forall \varphi_j \in V_{h,0}$$

Notons

$$A_\mu := \begin{pmatrix} a(\varphi_0, \varphi_0; \mu) \\ \vdots \\ a(\varphi_i, \varphi_j; \mu) \\ \vdots \\ a(\varphi_{Nel-1}, \varphi_j; \mu) \end{pmatrix} X := \begin{pmatrix} \alpha_0 \\ \vdots \\ \alpha_i \\ \vdots \\ \alpha_{Nel-1} \end{pmatrix} \text{ et } b := \begin{pmatrix} l(\varphi_0) \\ \vdots \\ l(\varphi_j) \\ \vdots \\ l(\varphi_{Nel-1}) \end{pmatrix}$$

Résoudre (4) revient à trouver X tel que

$$A_\mu X = b$$

où :

1. $A_\mu = A_1 + \mu A_2 + M$
2. $A_1 = \int_\Omega \mathbb{1}_{\Omega_1} \varphi'_i \varphi'_j$, $A_2 = \int_\Omega \mathbb{1}_{\Omega_2} \varphi'_i \varphi'_j$, $M = \int_\Omega \varphi_i \varphi_j$
3. $b = \int_\Omega f \varphi_j$

Après calculs on a :

$$(A_1)_{ij} = \begin{cases} a(\varphi_i, \varphi_j) = 0, x \notin [x_{i-1}, x_{i+1}] \\ a(\varphi_i, \varphi_{i+1}) = -\frac{1}{h}, x \in [x_i, x_{i+1}] \\ a(\varphi_i, \varphi_{i-1}) = -\frac{1}{h}, x \in [x_{i-1}, x_i] \\ a(\varphi_i, \varphi_i) = \frac{2}{h}, x \in [x_{i-1}, x_{i+1}] \end{cases}$$

De même pour A_2 on a :

$$(A_2)_{ij} = \begin{cases} a(\varphi_i, \varphi_j) = 0, x \notin [x_{i-1}, x_{i+1}] \\ a(\varphi_i, \varphi_{i+1}) = -\frac{1}{h}, x \in [x_i, x_{i+1}] \\ a(\varphi_i, \varphi_{i-1}) = -\frac{1}{h}, x \in [x_{i-1}, x_i] \\ a(\varphi_i, \varphi_i) = \frac{2}{h}, x \in [x_{i-1}, x_{i+1}] \end{cases}$$

Pour la matrice de masse M on a :

$$(M)_{ij} = \begin{cases} a(\varphi_i, \varphi_j) = 0, x \notin [x_{i-1}, x_{i+1}] \\ a(\varphi_i, \varphi_{i+1}) = \frac{h}{6}, x \in [x_i, x_{i+1}] \\ a(\varphi_i, \varphi_{i-1}) = \frac{h}{6}, x \in [x_{i-1}, x_i] \\ a(\varphi_i, \varphi_i) = \frac{2h}{3}, x \in [x_{i-1}, x_{i+1}] \end{cases}$$

Pour calculer le second membre, nous utilisons la méthode des trapèzes par morceaux.

$$b = \int_{\Omega} f \varphi_j = \sum_{j=0}^{Nel-1} \int_{x_j}^{x_{j+1}} f \varphi_j.$$

En supposant que $f = 1$, on a :

$$b_j = \int_{x_{j-1}}^{x_{j+1}} f \varphi_j = \frac{(h_j + h_{j+1})}{2}, x \in [x_{j-1}, x_{j+1}].$$

En fin, on obtient :

$$b_j = h, \forall j \in \{0, \dots, Nel - 1\}$$

Pour respecter les conditions aux bords, nous avons rendu les lignes associées aux degrés de liberté de Dirichlet, nulles, sauf sur la diagonale de A_{μ} la valeur 1. Pour finir les matrices se réécrivent :

$$b_j = \begin{cases} h, \forall j \in \{1, \dots, Nel - 2\} \\ 0, \text{ sinon} \end{cases} \quad (A_{\mu})_{ij} = \begin{cases} A_{ij}, \forall i, j \in \{1, \dots, Nel - 2\} \\ 1, \text{ lorsque } (i, j) = (0, 0) \text{ ou } (i, j) = (Nel-1, Nel-1) \\ 0, \text{ sinon} \end{cases}$$

Pour $\mu = 1$ et $Nel = 10$, on a :

$$A_1 = \begin{pmatrix} 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ -9 & 18 & -9 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & -9 & 18 & -9 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & -9 & 18 & -9 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & -9 & 18 & -9 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & -9 & 18 & -9 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & -9 & 18 & -9 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & -9 & 18 & -9 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \end{pmatrix}$$

$$A_2 = \begin{pmatrix} 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 9 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 9 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 9 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 9 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \end{pmatrix} \text{ et } b = \begin{pmatrix} 0 \\ 0.11 \\ 0.11 \\ 0.11 \\ 0.11 \\ 0.11 \\ 0.11 \\ 0.11 \\ 0.11 \\ 0 \end{pmatrix}$$

$$M = \begin{pmatrix} 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0.019 & 0.074 & 0.019 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0.019 & 0.074 & 0.019 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0.019 & 0.074 & 0.019 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0.019 & 0.074 & 0.019 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0.019 & 0.074 & 0.019 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0.019 & 0.074 & 0.019 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0.019 & 0.074 & 0.019 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0.019 & 0.074 & 0.019 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \end{pmatrix}$$

2.3 Validation

On vérifie que si $\mu = 1$ la solution éléments finis u_{EF} est cohérente avec la solution exacte u_{ex} . En effet, si $\mu = 1$, alors $D(x) \equiv 1$ et le système d'équations (1) devient :

$$\begin{cases} -u'' + u = 1 \\ u|_{\partial\Omega} = 0 \end{cases} \quad (5)$$

Ce nouveau problème (5) admet une unique solution $u_{ex}(x)$:

$$\forall x \in \Omega, u_{ex}(x) = C_1 \exp(-x) + C_2 \exp(x) + 1, \text{ avec } C_1 = \frac{1-e}{e-e^{-1}} \text{ et } C_2 = \frac{e^{-1}-1}{e-e^{-1}}$$

Pour vérifier la convergence de u_{EF} la solution approchée du problème (5) vers la solution exacte u_{ex} , nous avons les résultats ci-dessous qui permettent d'estimer l'erreur $u_{ex} - u_{EF}$ en norme $L^2(\Omega)$ et $H^1(\Omega)$.

Proposition 1 *On suppose que la solution du problème u est dans $H^2(\Omega)$ et on note u_h la solution du problème approché (5) Alors il existe une constante c dépendante seulement de Ω telle que*

$$\|u - u_h\|_{1,\Omega} \leq ch|u|_{2,\Omega}.$$

Proposition 2 *On note u la solution du problème et u_h la solution du problème approché (5) Alors il existe une constante c dépendante seulement de Ω telle que*

$$\|u - u_h\|_{0,\Omega} \leq ch^2|u|_{2,\Omega}.$$

Pour tout $u \in V_{h,0}$, on a :

$$\|u\|_{L^2(\Omega)}^2 = \int_{\Omega} u^2 dx = \sum_{i,j=1}^N u_i u_j \int_{\Omega} \varphi_i \varphi_j dx$$

$$\|u\|_{H^1(\Omega)}^2 = \int_{\Omega} u^2 + (u')^2 dx = \sum_{i,j=1}^N u_i u_j \int_{\Omega} \varphi_i \varphi_j dx + \sum_{i,j=1}^N u_i u_j \int_{\Omega} \varphi'_i \varphi'_j dx$$

En posant :

$$U := \begin{pmatrix} u_0 \\ \cdot \\ \cdot \\ \cdot \\ u_{Nel-1} \end{pmatrix} \quad N := A_1 + A_2$$

Finalement on en déduit :

$$\|u\|_{L^2(\Omega)}^2 = U^T N U \quad \text{et} \quad \|u\|_{H^1(\Omega)}^2 = U^T (M + N) U$$

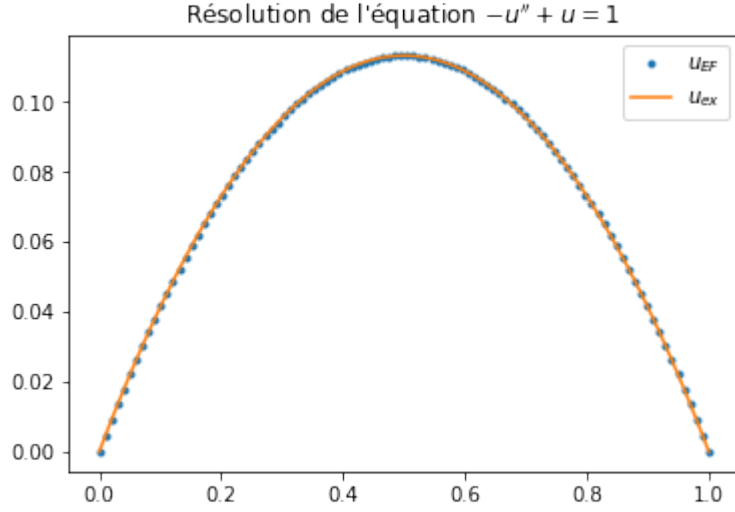


FIGURE 2 – Solution exacte u_{ex} et Solution élément fini u_{EF}

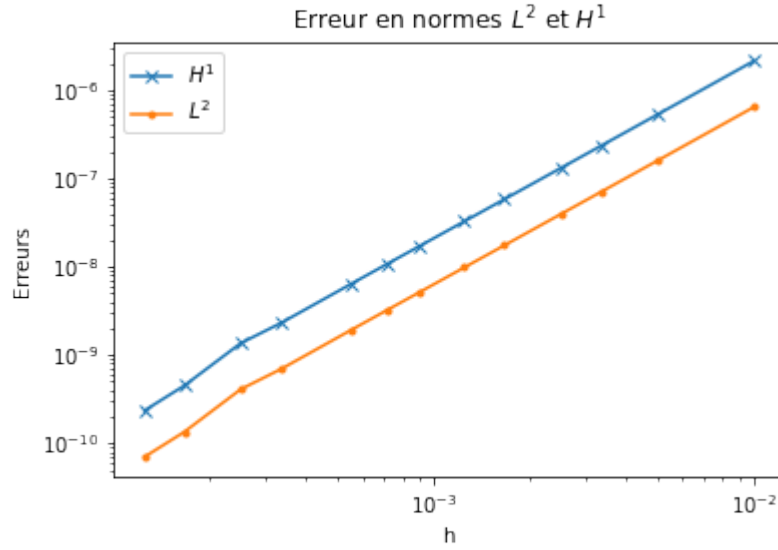


FIGURE 3 – Erreurs $\|u_{ex} - u_{EF}\|_{L^2}$ et $\|u_{ex} - u_{EF}\|_{H^1}$ en fonction de h

Pour différentes valeurs de N_{el} le nombre d'éléments du maillage dans $\{100, 200, 300, 400, 600, 800, 1100, 1400, 1800, 3000, 4000, 6000, 8000, 10000\}$, nous avons résolu le problème (5) et calculer l'erreur entre u_{ex} et u_{EF} en norme H^1 et L^2 . La pente de l'erreur est de 2 en norme H^1 et L^2 , ce qui est conforme aux attentes théoriques.

3 Modèle réduit

3.1 Formulation du problème réduit

La méthode des bases réduites s'appuie sur le fait que l'évolution de la solution de l'EDP $u(\mu)$ n'occupe pas tout l'espace $H_0^1(\Omega)$ dans lequel elle est recherchée, mais seulement une variété V ayant une dimension plus faible. Dans notre cas, la variété V est définie par :

$$V = \{u(\mu) \in H_0^1 | \mu \in P_M\}$$

De la même manière que nous approchons l'espace $H_0^1(\Omega)$ par l'espace discret V_h , nous construisons une approximation V_{N_0} de V de la manière suivante :

$$V_{N_0} = \{u(\mu) \in V_h | \mu \in P_M\}$$

La méthode des bases réduites repose sur une projection de type Galerkin sur un espace V_{N_0} de dimension N_0 , avec $N_0 \ll \dim(V_h)$. Nous verrons comment construire l'espace V_{N_0} afin qu'il approche au mieux de V_h .

En remplaçant V_h par V_{N_0} dans le problème approché (4), le problème réduit s'écrit :

$$\text{Trouver } u_\mu^{BR} \in V_{N_0} \text{ tel que } a_\mu(u_\mu^{BR}, u_i) = b(u_i), \forall i \in \{1, \dots, N_0\} \quad (6)$$

3.2 Espace des bases réduites V_{N_0}

Pour commencer, introduisons l'ensemble de paramètres $P_M := \{\mu_1, \dots, \mu_M\}$. Cet ensemble, de cardinal M , a des valeurs strictement positives. Ensuite nous définissons l'ensemble P_{N_0}

$$P_{N_0} := \{\mu_1, \dots, \mu_{N_0}\} \text{ avec } N_0 \ll M$$

à laquelle on associe l'espace des bases réduites

$$V_{N_0} := \text{vect}(u_1, u_2, \dots, u_{N_0})$$

Les fonctions u_i pour $i \in P_{N_0}$ sont appelés « snapshot ». Elles sont obtenues en résolvant le problème approché par éléments finis. La construction de l'espace réduit V_{N_0} , qui approche au mieux V_h , est faite à l'aide de l'algorithme glouton dont une implémentation est donnée ci-dessous.

Algorithm 1 Algorithme Glouton

Require: P ensemble, $U^{ef}(\mu)$ et $U^{br}(\mu)$ 2 fonctions

choisir μ_1 de manière aléatoire

$u(\mu_1) = u_1$

$B = Vect(u(\mu_1))$

for i allant de 2 à N_0 **to** ... **do**

$u_i := \underset{\mu \in P}{argmax} (||u(\mu) - P_{B_i}u(\mu)||)$

$B_i := Vect(u_1, \dots, u_i)$

end for

Ensure: retourner B_{N_0}

3.3 Matrices réduites

On sait que u_μ^{BR} et u_i se décomposent sous la forme

$$u_{BR}^\mu = \sum_{i=1}^{N_0} X_{\mu,i}^{BR} u_i \text{ et } u_i = \beta_0 \varphi_0 + \dots + \beta_{Nel-1} \varphi_{Nel-1}$$

Et par bilinéaire et linéarité de l et a_μ , on a :

$$a_\mu(u_\mu^{BR}, u_i) = \sum_{i=1}^{N_0} \sum_{k,j=0}^{Nel-1} X_{\mu,i}^{BR} \beta_{i,k} \beta_{i,j} a_\mu(\varphi_k, \varphi_j)$$

$$l(u_i) = \sum_{i=1}^{N_0} \sum_{j=0}^{Nel-1} \beta_{i,j} l(\varphi_j)$$

Posons maintenant

$$X^{BR} = \begin{pmatrix} X_{\mu,1}^{BR} \\ \vdots \\ X_{\mu,N_0}^{BR} \end{pmatrix} U = (u_1, \dots, u_{N_0})$$

On réécrit l'équation (6) sous forme matricielle :

$$(U^T A_\mu U) X^{BR} = U^T (A_0 + M) U + \mu U^T A_1 U = U^T b$$

Notons

$$V_0^{BR} := U^T (A_0 + M) U \text{ et } V_1^{BR} := U^T A_1 U$$

Ainsi on a les relations suivantes :

$$B^{BR} := U^T b \text{ et } A_\mu^{BR} = V_0^{BR} + \mu V_1^{BR}$$

Résoudre le problème réduit (6) revient finalement à trouver X^{BR} tel que :

$$A_{\mu}^{BR} X^{BR} = B^{BR}$$

La méthode des bases réduites se décompose en deux étapes : «hors-ligne» et «en-ligne». À l'étape hors-ligne, le vecteur B^{BR} et les matrices V_0^{BR} et V_1^{BR} sont calculées une seule fois. À l'étape en-ligne, pour un certain $\mu \in P_M$, on assemble la matrice A_{μ}^{BR} et resout le système réduit

$$A_{\mu}^{BR} X^{BR} = B^{BR}$$

3.4 Etude du modèle réduit

Dans cette section, nous allons analyser la solution réduite du problème (5). Pour ce faire, il faut d'abord choisir la valeur N_0 afin d'appliquer l'algorithme glouton.

Après la construction de V_0^{BR} et V_1^{BR} pour diverses valeurs de N_0 , nous avons calculé leur conditionnement respectif. Quand N_0 est supérieur ou égale à 5, le problème devient mal conditionné et leurs conditionnements sont alors de l'ordre de 10^{16} d'après le graphique (3.4). Lorsque N_0 est égal à 2, le résultat est très insuffisant. Les valeurs convenables de N_0 sont donc 3,4,5. Dans la suite, nous fixons N_0 à 3 pour avoir un conditionnement pas trop élevé.

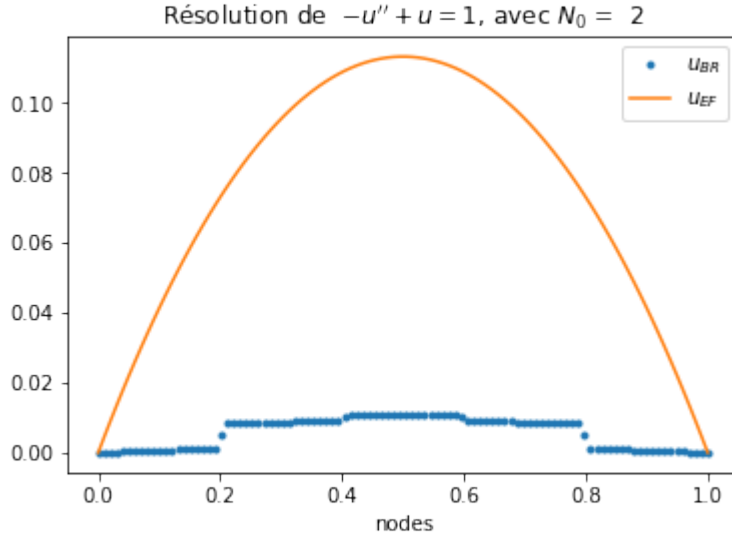


FIGURE 4 – Solution Elément Finis et Base Réduite

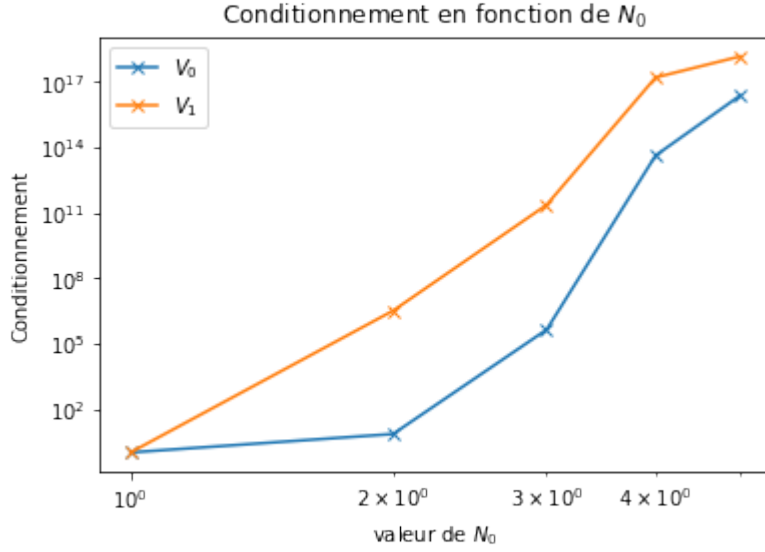


FIGURE 5 – Evolution du conditionnement

Vérifions maintenant que V_3^{BR} est une base stable, en calculant l'erreur $\epsilon := \max_{\mu \in P_M} \|u^{EF}(\mu) - u^{RB}(\mu)\|_{L^2}$ avec M dans $\{10, 100, 500, 1000, 2000, 2500, 3000\}$. Après calculs, la valeur de ϵ est très petite de l'ordre de 10^{-20} et constante. Comme l'erreur ϵ varie peu, on peut dire que l'algorithme Glouton construit une base V_3^{BR} convenable.

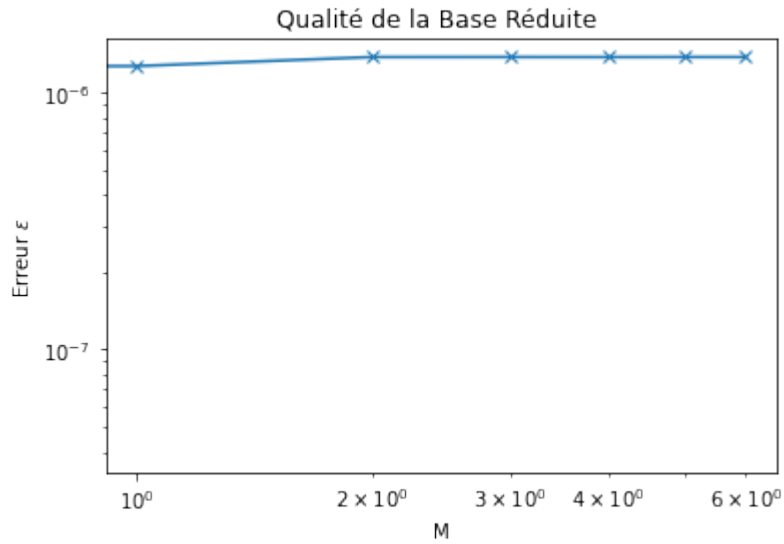


FIGURE 6 – Erreur ϵ pour différent valeur de M

Maintenant on souhaite savoir sur quelles parties de Ω le modèle base réduit est le moins

performant. En fixant μ à 0.05 et N_{el} à 100, l'erreur absolue entre la solution éléments finis et base réduite est d'ordre 10^{-3} dans les intervalles $[0.4, 0.6]$, $[0.2, 0.4]$ et $[0.6, 0.8]$ d'après le graphique ci-dessous (3.4).

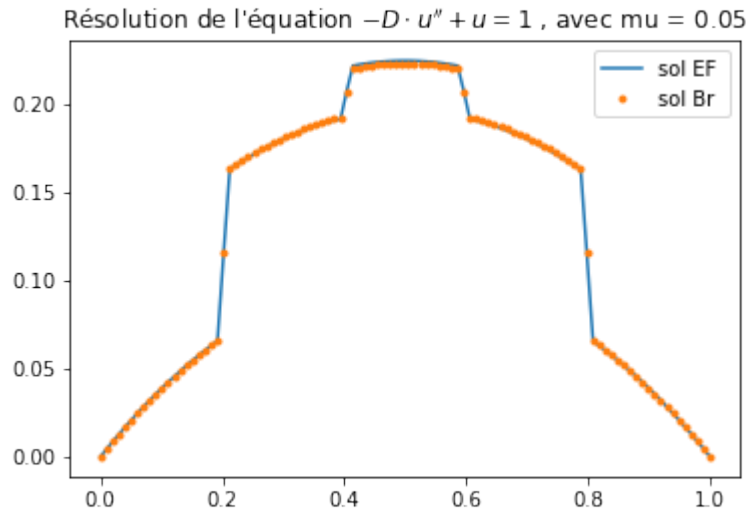


FIGURE 7 – Solution Élément Finis et Base Réduite

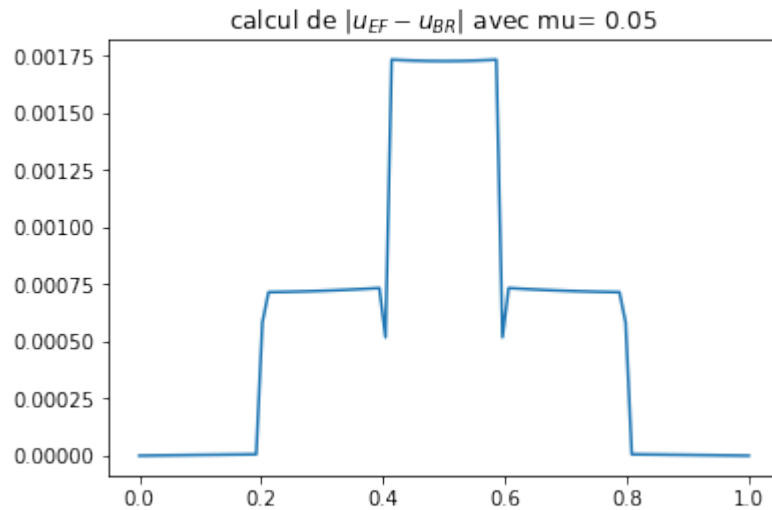


FIGURE 8 – Erreur en valeur absolu entre les solutions EF et BR

Références

- [1] Gianluigi Rozza, *An introduction to reduced basis method for parametrized PDEs*

- [2] B. Haasdonk, *Reduced Basis Methods for Parametrized PDEs – A Tutorial Introduction for Stationary and Instationary Problems*
- [3] Alexandre Ern, *Analyse numérique et optimisation Méthode des bases réduites*
- [4] Gwenol Grandperrin, *Introduction à la méthode des bases réduites*