



ÉCOLE POLYTECHNIQUE
FÉDÉRALE DE LAUSANNE

Tests de méthodes numériques de la bibliothèque éléments finis LifeV: comparaison avec la théorie

Céline CHATELAIN
Sabina SCHNEIDER

Sous la direction de
Prof. Alfio Quarteroni et Dr. Christophe Prud'Homme

12 février 2005

Table des matières

Introduction	6
1 Théorie	8
1.1 Passage de la forme forte à la forme faible	9
1.1.1 Le problème	9
1.1.2 Obtention de la forme faible	9
1.2 Approximation par éléments finis : problème de Galerkin . . .	10
1.2.1 Discrétisation	10
1.2.2 Analyse du problème discret	11
1.2.3 Estimation de l'erreur d'approximation	11
1.3 Problème en dimension $d > 1$	12
1.4 Normes et seminormes	13
2 Présentation du problème	15
2.1 Le problème $x^2y^2z^2$	15
2.2 Le problème <i>sincosd</i>	16
3 Programmation	20
3.1 Description des programmes	20
3.2 Design pattern	22
4 Résultats	23
4.1 Le problème $x^2y^2z^2$	23
4.2 Le problème <i>sincosd</i>	28
5 Conclusion	32
A Résultats pour la fonction $x^2y^2z^2$ en \mathbb{P}^1	34
B Résultats pour la fonction $x^2y^2z^2$ en \mathbb{P}^2	38
C Résultats pour la fonction <i>sincosd</i> en \mathbb{P}^1	41
D Résultats pour la fonction <i>sincosd</i> en \mathbb{P}^2	44

E Programmes	47
E.1 errorEvaluator	47
E.2 statistics	52
E.3 ud_functions	57
E.4 test_scalar_fe_convergence	62
E.5 Maillage	74
E.6 draw_function	78
Bibliographie	79

Table des figures

2.1	Maillage du cube pour $h = \frac{1}{4}$	19
2.2	Maillage du cube pour $h = \frac{1}{16}$	19
4.1	Graphique de la fonction $x^2y^2z^2$ pour $\beta = -10$ en \mathbb{P}^1	26
4.2	Graphique de la fonction $x^2y^2z^2$ pour $\beta = 0$ en \mathbb{P}^1	26
4.3	Graphique de la fonction $x^2y^2z^2$ pour $\beta = 1000$ en \mathbb{P}^1	26
4.4	Graphique de la fonction $x^2y^2z^2$ pour $\beta = -10$ en \mathbb{P}^2	27
4.5	Graphique de la fonction $x^2y^2z^2$ pour $\beta = 0$ en \mathbb{P}^2	27
4.6	Graphique de la fonction $x^2y^2z^2$ pour $\beta = 1000$ en \mathbb{P}^2	27
4.7	Graphique de la fonction sincosd pour $\beta = -10$ en \mathbb{P}^1	30
4.8	Graphique de la fonction sincosd pour $\beta = 0$ en \mathbb{P}^1	30
4.9	Graphique de la fonction sincosd pour $\beta = 1000$ en \mathbb{P}^1	30
4.10	Graphique de la fonction sincosd pour $\beta = -10$ en \mathbb{P}^2	31
4.11	Graphique de la fonction sincosd pour $\beta = 0$ en \mathbb{P}^2	31
4.12	Graphique de la fonction sincosd pour $\beta = 1000$ en \mathbb{P}^2	31

Liste des tableaux

4.1	Résultats pour le problème $x2y2z2$ en \mathbb{P}^1 avec $\beta = -10$. . .	25
4.2	Résultats pour le problème $x2y2z2$ en \mathbb{P}^1 avec $\beta = 0$	25
4.3	Résultats pour le problème $x2y2z2$ en \mathbb{P}^1 avec $\beta = 1000$. . .	25
4.4	Résultats pour le problème $x2y2z2$ en \mathbb{P}^2 avec $\beta = -10$. . .	25
4.5	Résultats pour le problème $x2y2z2$ en \mathbb{P}^2 avec $\beta = 0$	25
4.6	Résultats pour le problème $x2y2z2$ en \mathbb{P}^2 avec $\beta = 1000$. . .	25
4.7	Résultats pour le problème $sincosd$ en \mathbb{P}^1 avec $\beta = -10$. . .	29
4.8	Résultats pour le problème $sincosd$ en \mathbb{P}^1 avec $\beta = 0$	29
4.9	Résultats pour le problème $sincosd$ en \mathbb{P}^1 avec $\beta = 1000$. . .	29
4.10	Résultats pour le problème $sincosd$ en \mathbb{P}^2 avec $\beta = -10$. . .	29
4.11	Résultats pour le problème $sincosd$ en \mathbb{P}^2 avec $\beta = 0$	29
4.12	Résultats pour le problème $sincosd$ en \mathbb{P}^2 avec $\beta = 1000$. . .	29
A.1	Norme de l'énergie, $\beta = -10$	34
A.2	Norme de l'énergie, $\beta = 0$	34
A.3	Norme de l'énergie, $\beta = 1000$, pente= 1.4323	34
A.4	Norme H^1 , $\beta = -10$,pente=1.2959	35
A.5	Norme H^1 , $\beta = 0$, pente=1.21152	35
A.6	Norme H^1 , $\beta = 1000$, pente= 1.32231	35
A.7	Norme L^1 , $\beta = -10$	35
A.8	Norme L^1 , $\beta = 0$	35
A.9	Norme L^1 , $\beta = 1000$, pente= 1.93947	36
A.10	Norme L^2 , $\beta = -10$, pente=1.90626	36
A.11	Norme L^2 , $\beta = 0$, pente= 1.99776	36
A.12	Norme L^2 , $\beta = 1000$, pente= 1.97218	36
A.13	Norme L^∞ , $\beta = -10$	36
A.14	Norme L^∞ , $\beta = 0$	37
A.15	Norme L^∞ , $\beta = 1000$	37
B.1	Norme de l'énergie, $\beta = -10$, pente= -5.24867	38
B.2	Norme de l'énergie, $\beta = 0$, pente= -3.14393	38
B.3	Norme de l'énergie, $\beta = 1000$, pente= -2.04915	38
B.4	Norme H^1 , $\beta = -10$, pente= -5.33914	39
B.5	Norme H^1 , $\beta = 0$, pente= -3.15131	39

B.6	Norme H^1 , $\beta = 1000$, pente= -2.1108	39
B.7	Norme L^1 , $\beta = -10$, pente= -5.71966	39
B.8	Norme L^1 , $\beta = 0$, pente= -3.63878	39
B.9	Norme L^1 , $\beta = 1000$, pente= -2.15851	39
B.10	Norme L^2 , $\beta = -10$, pente= -5.83913	40
B.11	Norme L^2 , $\beta = 0$, pente= -3.6937	40
B.12	Norme L^2 , $\beta = 1000$, pente= -2.2064	40
B.13	Norme L^∞ , $\beta = -10$	40
B.14	Norme L^∞ , $\beta = 0$	40
B.15	Norme L^∞ , $\beta = 1000$	40
C.1	Norme de l'énergie, $\beta = -10$, pente= 0.707029	41
C.2	Norme de l'énergie, $\beta = 0$, pente= 0.697586	41
C.3	Norme de l'énergie, $\beta = 1000$, pente= 1.25561	41
C.4	Norme H^1 , $\beta = -10$, pente= 0.742941	42
C.5	Norme H^1 , $\beta = 0$, pente= 0.715065	42
C.6	Norme H^1 , $\beta = 1000$, pente= 0.457198	42
C.7	Norme L^1 , $\beta = -10$, pente= 1.16635	42
C.8	Norme L^1 , $\beta = 0$, pente= 1.39534	42
C.9	Norme L^1 , $\beta = 1000$, pente= 1.6804	42
C.10	Norme L^2 , $\beta = -10$, pente= 1.23749	43
C.11	Norme L^2 , $\beta = 0$, pente= 1.29431	43
C.12	Norme L^2 , $\beta = 1000$, pente= 1.5278	43
C.13	Norme L^∞ , $\beta = -10$	43
C.14	Norme L^∞ , $\beta = 0$	43
C.15	Norme L^∞ , $\beta = 1000$	43
D.1	Norme de l'énergie, $\beta = -10$, pente= 0.717731	44
D.2	Norme de l'énergie, $\beta = 0$, pente= 0.710636	44
D.3	Norme de l'énergie, $\beta = 1000$, pente= 0.64142	44
D.4	Norme H^1 , $\beta = -10$, pente= 0.743898	45
D.5	Norme H^1 , $\beta = 0$, pente= 0.71998	45
D.6	Norme H^1 , $\beta = 1000$, pente= 0.560778	45
D.7	Norme L^1 , $\beta = -10$, pente= 1.7957	45
D.8	Norme L^1 , $\beta = 0$, pente= 1.79943	45
D.9	Norme L^1 , $\beta = 1000$, pente= 1.97514	45
D.10	Norme L^2 , $\beta = -10$, pente= 1.62183	46
D.11	Norme L^2 , $\beta = 0$, pente= 1.59841	46
D.12	Norme L^2 , $\beta = 1000$, pente= 1.42308	46
D.13	Norme L^∞ , $\beta = -10$	46
D.14	Norme L^∞ , $\beta = 0$	46
D.15	Norme L^∞ , $\beta = 1000$	46

Introduction

Nous travaillons au sein du projet LifeV. Ce projet regroupe un grand nombre de chercheurs qui développent ensemble une implémentation des méthodes de résolution d'Équations aux Dérivées Partielles(EDPs) par éléments finis.

Notre rôle consiste à créer des tests de convergence automatiques pour les méthodes développées par les chercheurs de LifeV de façon à détecter d'éventuelles erreurs survenues lors de l'implémentation. Nous utiliserons des résultats théoriques pour créer nos tests, afin de découvrir d'éventuelles erreurs dans le code.

L'intérêt de ce projet se trouve dans le contrôle systématique des méthodes implémentées dans le code LifeV et ainsi du gain de temps et d'énergie, non utilisés à la recherche de ces erreurs sans les tests que nous allons implémenter.

Déroulement du travail

Notre travail commence par un apprentissage de l'environnement LifeV et de la théorie relative aux EDPs. Nous apprenons cette théorie pour une dimension et généralisons directement à trois dimensions pour l'application pratique. Nous verrons que cette généralisation est tout à fait évidente, mais indispensable.

Une très grande partie du travail est consacrée à la programmation. Nous commençons par implémenter quelques classes, dont nous aurons besoin par la suite. Ces programmes permettront par exemple d'évaluer une fonction en une valeur donnée. Ce travail nous permet surtout de nous familiariser avec l'environnement et de mettre à jour nos connaissances en programmation.

Ensuite, nous nous lancerons effectivement dans le travail demandé : nous analyserons deux problèmes classiques (chapitre 2). Pour ce faire, nous devons implémenter une classe *sincosd* et modifier l'environnement de façon à pouvoir l'utiliser pour nos tests. Nous utiliserons alors nos programmes du début pour l'analyse numérique et graphique des résultats. Ceci nous permettra de les comparer directement aux résultats attendus par la théorie.

Nous apprendrons aussi à utiliser de nouveaux outils comme les *pattern*

design, fréquemment utilisés pour améliorer la structure du code (afin de le rendre plus consistant et réutilisable, cf. chapitre 3).

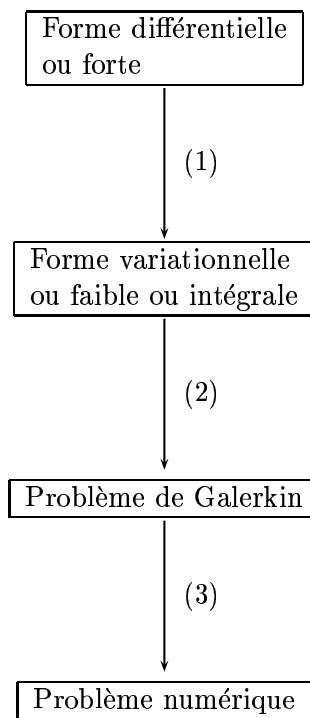
Le lecteur intéressé trouvera par ailleurs en annexe toutes les valeurs numériques trouvées, ainsi que le code que nous aurons implémenté.

Chapitre 1

Théorie

Notre projet nécessite une connaissance en éléments d'approximation des Équations aux Dérivées Partielles (EDPs). Nous allons présenter la résolution par la méthode des éléments finis (MEF).

La résolution par la méthode des éléments finis peut être schématisée comme suit :



Dans les paragraphes qui suivent, nous allons vous expliquer comment se font les passages (1), (2) et (3). D'abord dans le cas monodimensionnel, puis dans le cas général.

1.1 Passage de la forme forte à la forme faible

1.1.1 Le problème

Tout les problèmes que nous allons analyser s'écrivent sous la forme différentielle suivante :

$$\begin{aligned} & \text{chercher } u : [a, b] \longrightarrow \mathbb{R}, \text{ tq.} \\ & \left\{ \begin{array}{l} Lu = f \quad \text{dans } (a, b) \\ + \text{ c.b.} \end{array} \right. \quad (1.1) \end{aligned}$$

Remarque 1.1. Les conditions au bord (c.b.) peuvent être données sous différentes formes :

- conditions de Dirichlet : $u(a) = u_a, u(b) = u_b$ avec u_a, u_b des valeurs données ;
- conditions de Neumann : $u'(a) = \phi_a, u'(b) = \phi_b$ avec ϕ_a, ϕ_b des valeurs données ;
- conditions mixtes : mélange des deux précédentes ;

Si $u_a = u_b = 0$ ou $\phi_a = \phi_b = 0$, on parle de conditions homogènes.

On appelle *solution classique (ou forte)* de (1.1) toute fonction $u \in \mathcal{C}^2[a, b]$ satisfaisant (1.1) en chaque point de (a, b) . Il existe cependant des situations où on cherche des solutions « non-classiques ». Afin de donner un sens à ces dernières, on reformule le problème sous une forme variationnelle :

$$\left\{ \begin{array}{l} \text{chercher } u \in V, \text{ t.q.} \\ a(u, v) = F(v) \quad \forall v \in V \end{array} \right. \quad (1.2)$$

1.1.2 Obtention de la forme faible

Cette forme variationnelle abstraite se trouve comme suit :

1. on suppose que $f \in \mathcal{C}^0[a, b]$ et que u est solution régulière de classe \mathcal{C}^2 de (1.1) ;
2. on choisit $v \in \mathcal{C}^1[a, b]$, une fonction régulière dite *fonction test* et on multiplie (1.1)₁ par v ;
3. on intègre par partie dans le terme d'ordre 2 ;
4. on prend f, u et v dans des espaces de faible régularité pour lesquels les intégrales trouvées en 3 existent. De plus, on choisit V de telle sorte que $u, v \in V$ satisfont aux conditions de bord ;
5. on peut ainsi définir une forme bilinéaire

$$\begin{aligned} a : V \times V & \longrightarrow \mathbb{R} \\ (u, v) & \longmapsto a(u, v) \end{aligned}$$

et une fonctionnelle linéaire

$$F : V \longrightarrow \mathbb{R}$$

$$v \longmapsto F(v) = \int_a^b f v.$$

Toute solution u de (1.2) est une *solution faible*. On peut montrer que si u est solution classique, alors u est solution faible; et réciproquement, toute solution faible régulière est solution forte.

Le théorème suivant donne les conditions d'existence et d'unicité de la solution faible :

Théorème 1.1 (Lax-Milgram). *Soit*

1. V un espace de Hilbert
2. $a : V \times V \longrightarrow \mathbb{R}$ une forme bilinéaire, continue et coercive
3. $F : V \longrightarrow \mathbb{R}$ une forme linéaire continue.

Alors le problème variationnel (1.2) admet une solution unique $u \in V$.

1.2 Approximation par éléments finis : problème de Galerkin

1.2.1 Discrétisation

La méthode des éléments finis est une méthode de Galerkin. L'idée de base consiste à remplacer dans la forme faible (1.2) l'espace de dimension infinie V par un espace de dimension finie $V_h \subset V$ tel que $\dim(V_h) = N_h \rightarrow \infty$ lorsque $h \rightarrow 0$. On obtient alors le problème suivant, appelé *problème discret* :

$$\begin{cases} \text{chercher } u_h \in V_h, \text{ t.q.} \\ a(u_h, v_h) = F(v_h) \quad \forall v_h \in V_h \end{cases} \quad (1.3)$$

Ce problème étant de dimension finie, on peut définir une base $\{\varphi_i\}_{i=1}^N$ de V_h avec $N = N_h$. On peut alors écrire

$$u_h(x) = \sum_{i=1}^N u_j \varphi_j(x)$$

Et par la linéarité du problème, il suffit de prendre $v_h = \varphi_i$ pour $i = 1, \dots, N$. Le problème discret (1.3) est alors équivalent à

$$\begin{cases} \text{trouver } \{u_j\}_{j=1}^N, \text{ t.q.} \\ \sum_{j=1}^{N_h} u_j a(\varphi_j, \varphi_i) = F(\varphi_i) \quad \forall i = 1, \dots, N \end{cases} \quad (1.4)$$

On peut introduire la matrice de raideur $A = (a_{ij})$ tel que $a_{ij} = a(\varphi_j, \varphi_i)$, le vecteur composé des inconnues $u = (u_1, \dots, u_N)$ et le second membre $g = (g_1, \dots, g_N)$ avec $g_i = F(\varphi_i)$. Et on obtient le système linéaire

$$Au = g$$

1.2.2 Analyse du problème discret

On introduit une partition \mathcal{T}_h de l'intervalle $[a, b]$ en n sous-intervalles $I_j = [x_j, x_{j+1}]$ de longueur h_j pour $j = 1, \dots, n-1$ avec $x_0 = a$, $x_n = b$, $x_i < x_j$ pour tout $i < j$ et $h = \max_j h_j$. On introduit ainsi l'espace des éléments finis de Lagrange de degré $r \geq 1$:

$$X_h^r = \{v_h \in \mathcal{C}^0[a, b] \mid v_h|_{[x_j, x_{j+1}]} \in \mathbb{P}^r, j = 0, \dots, n-1\}$$

On peut maintenant poser $V_h = X_h^r \cap V$.

Si le problème variationnel vérifie les conditions du théorème de Lax-Milgram (théorème 1.1) et donc admet une solution unique, alors le problème de Galerkin aussi (car $V_h \subset V$). On peut montrer que cette méthode est stable.

Le théorème suivant nous permet de montrer la convergence de u_h vers u lorsque $h \rightarrow 0$:

Théorème 1.2 (Lemme de Céa). *Soit V un espace de Hilbert, $F \in V'$ et $a : V \times V \rightarrow \mathbb{R}$ bilinéaire, continue et coercive. Alors*

$$\|u - u_h\| \leq \frac{M}{\alpha} \inf_{v_h \in V_h} \|u - v_h\|_V$$

avec u la solution du problème fort (1.1), u_h la solution du problème de Galerkin (1.4) et M, α les constantes de continuité et coercivité de a .

Corollaire 1.3 (Convergence de la méthode de Galerkin). *Si la famille de sous-espaces V_h satisfait*

$$\lim_{h \rightarrow 0} \inf_{v_h \in V_h} \|v - v_h\|_V = 0 \quad \forall v \in V$$

et sous les hypothèses du lemme de Céa, on a

$$\lim_{h \rightarrow 0} \|u - u_h\|_V = 0$$

1.2.3 Estimation de l'erreur d'approximation

On introduit l'opérateur d'interpolation

$$\Pi_h^r : \mathcal{C}^0[a, b] \rightarrow X_h^r$$

où pour chaque $v \in \mathcal{C}^0[a, b]$, $\Pi_h^r v \in X_h^r$ est l'interpolant de v dans X_h^r défini par

$$\Pi_h^r v(\xi) = v(\xi)$$

pour tout point ξ associé aux degrés de liberté des fonctions de X_h^r .

L'erreur d'approximation est $\|u - u_h\|_V$ et l'erreur d'interpolation est $\|u - \Pi_h^r u\|_V$. Cette deuxième erreur est suffisante pour estimer la première :

Théorème 1.4. *Pour $k = 0, 1$ et $r \geq 1$, $\exists C_{k,r} > 0$, une constante telle que*

$$\|v - \Pi_h^r v\|_{H^k(a,b)} \leq C_{k,r} h^{r+1-k} |v|_{H^{r+1}(a,b)}, \quad \forall v \in H^{r+1}(a,b)$$

Corollaire 1.5. *Si $u \in H^{r+1}(a,b) \cap V$, $\exists C > 0$ constante, telle que*

$$\|u - u_h\|_V \leq Ch^r |u|_{H^{r+1}(a,b)}$$

On obtient donc que l'erreur d'approximation est de l'ordre de h^r , pourvu que la solution u soit suffisamment régulière.

1.3 Problème en dimension $d > 1$

En plusieurs dimensions, le problème se résout essentiellement de la même manière que pour le cas monodimensionnel.

Le problème fort s'écrit alors :

chercher $u : \Omega \rightarrow \mathbb{R}$, tq.

$$\begin{cases} Lu = f & \text{dans } \Omega \subset \mathbb{R}^d \\ + \text{c.b.} \end{cases} \quad (1.5)$$

Le passage à la forme faible est très semblable à la construction monodimensionnelle. Il suffit en effet de remplacer l'intégration par partie du terme d'ordre 2, par la formule de Green qui est plus générale. Le reste ne change pas.

Définition 1.1. La formule de Green est donnée par

$$\int_{\Omega} \nabla^2 u \cdot v = \int_{\partial\Omega} \frac{\partial u}{\partial n} \cdot v - \int_{\Omega} \nabla u \cdot \nabla v$$

où n est le vecteur normal à la surface u .

De la même manière qu'en dimension 1, on peut discrétiser le problème (1.5) et aboutir au système $Au = f$ qu'on va résoudre par un solveur du type *gradient conjugué*.

En remplaçant la partition de l'intervalle $[a, b]$ par une triangulation du domaine Ω , on peut généraliser l'analyse de convergence et l'estimation de l'erreur d'approximation.

1.4 Normes et seminormes

Pour pouvoir travailler convenablement avec les espaces évoqués ci-dessus, nous allons les définir et leur attribuer leur norme respective. Mais, d'abord nous avons besoin de la définition suivante :

Définition 1.2. On appelle notation multi-indice pour les dérivées de f ,

$$D^\alpha f(x) = \frac{\partial^{|\alpha|} f(x)}{\partial x_1^{\alpha_1} \cdots \partial x_d^{\alpha_d}}$$

où $\alpha = (\alpha_1, \dots, \alpha_d) \in \mathbb{N}^d$ avec $|\alpha| = \alpha_1 + \dots + \alpha_d$

On définit maintenant les espaces de faible régularité les plus utilisés :

$$L^p(\Omega) = \{f : \Omega \rightarrow \mathbb{R} \mid f \text{ - Lebesgue mesurable et } \int_{\Omega} |f(x)|^p < \infty\}$$

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

$$H_0^1(\Omega) = \{f \in H^1(\Omega) \mid f = 0 \text{ sur } \partial\Omega\}$$

Les normes et semi-normes associées sont :

$$\begin{aligned} \|e\|_{L^p(\Omega)} &= \left(\int_{\Omega} |e|^p \right)^{\frac{1}{p}} \\ \|e\|_{H^k(\Omega)} &= \sqrt{\sum_{|\alpha| \leq k} \int_{\Omega} |D^\alpha e|^2} \\ \|e\|_{H^k(a,b)} &= \sqrt{\sum_{|\alpha|=k} \int_{\Omega} |D^\alpha e|^2} \end{aligned}$$

En particulier, nous utiliserons régulièrement les normes suivantes :

$$\|e\|_{L^1(\Omega)} = \int_{\Omega} |e| \tag{1.6}$$

$$\|e\|_{L^2(\Omega)} = \sqrt{\int_{\Omega} e^2} \tag{1.7}$$

$$\|e\|_{L^\infty(\Omega)} = \sup\{|e(x)| \mid x \in \Omega\} \tag{1.8}$$

$$\|e\|_{H^1(\Omega)} = \sqrt{\int_{\Omega} (e^2 + \nabla e \nabla e)} \tag{1.9}$$

$$\|e\|_a = \sqrt{a(e, e)} \text{ est la norme d'énergie} \tag{1.10}$$

Ainsi que les semi-normes :

$$|v|_{H^1(a,b)} = \sqrt{\sum_{r=1}^d \int_{\Omega} \left(\frac{\partial u}{\partial x_r} \right)^2} \quad (1.11)$$

$$|v|_{H^2(a,b)} = \sqrt{\sum_{r=1}^d \sum_{s=1}^d \int_{\Omega} \left(\frac{\partial^2 u}{\partial x_r \partial x_s} \right)^2} \quad (1.12)$$

Chapitre 2

Présentation du problème

Nous allons considérer deux problèmes modèles. Le premier est caractérisé par un u polynômial de degré 2 et nous verrons que cela pose un problème pour l'estimation par MEF d'ordre 2. Alors, nous prendrons un deuxième problème non polynômial. Nous allons aussi considérer trois valeurs pour β (-10, 0 et 1000).

2.1 Le problème $x^2y^2z^2$

Le premier problème est donné par :

$$\begin{aligned} & \text{chercher } u : \Omega \longrightarrow \mathbb{R}, \text{ tq.} \\ & \begin{cases} -\nabla^2 u + \beta u = f & \text{dans } \Omega = [0, 1]^3 \\ u = g & \text{sur } \partial\Omega \end{cases} \end{aligned} \quad (2.1)$$

avec conditions aux bords de Dirichlet

$$g(x, y, z) = (x - 0.5)^2 + (y - 0.5)^2 + (z - 0.5)^2 \quad (2.2)$$

et solution exacte

$$u(x, y, z) = (x - 0.5)^2 + (y - 0.5)^2 + (z - 0.5)^2 \quad (2.3)$$

et on trouve

$$\begin{aligned} f : \Omega & \longrightarrow \mathbb{R} \\ x & \longmapsto -6 + \beta u \end{aligned} \quad (2.4)$$

On cherche la formulation faible de (2.1) :

Supposons que u est solution régulière de classe 2 du problème (2.1). Soit $v \in \mathcal{C}^1(\Omega)$ une fonction test, on multiplie l'équation de départ par v et on intègre sur Ω :

$$\int_{\Omega} -\nabla^2 u \cdot v + \int_{\Omega} \beta uv = \int_{\Omega} f v$$

on applique la formule de Green sur le terme d'ordre 2 :

$$\int_{\Omega} \nabla u \cdot \nabla v + \int_{\Omega} \beta uv - \int_{\partial\Omega} \frac{\partial u}{\partial n} \cdot v = \int_{\Omega} f v \quad \forall v \in V$$

La formulation faible est maintenant obtenue à partir de (2.1) en prenant des espaces de faible régularité, mais pour lesquels les intégrales existent. Ceci nous amène à prendre

$$u, v \in V = H^1(\Omega)$$

Par conséquent, d'après (2.1), la forme faible s'écrit :

$$\begin{cases} \text{chercher } u \in V, \text{ tq.} \\ a(u, v) = F(v) \quad \forall v \in V \end{cases} \quad (2.5)$$

où

$$\begin{aligned} a : V \times V &\longrightarrow \mathbb{R} \\ a(u, v) &= \int_{\Omega} \nabla u \cdot \nabla v + \int_{\Omega} \beta uv - \int_{\partial\Omega} \frac{\partial u}{\partial n} \cdot v \\ F : V &\longrightarrow \mathbb{R} \\ F(v) &= \int_{\Omega} f v \end{aligned}$$

2.2 Le problème *sincosd*

On étudie maintenant le deuxième problème :

$$\begin{cases} -\nabla^2 u + \beta u = f & \text{dans } \Omega = [0, 1]^d \\ u = 0 & x_1 = 0, x_1 = 1 \\ \frac{\partial u}{\partial n} = 0 & x_k = 0, x_k = 1, k = 2, \dots, d \end{cases} \quad (2.6)$$

avec solution exacte

$$u(x_1, x_2, \dots, x_d) = \sin(\pi x_1) \prod_{j=2}^d \cos(\pi x_j) \quad (2.7)$$

et on trouve

$$\begin{aligned} f : \Omega &\longrightarrow \mathbb{R} \\ x &\longmapsto (\pi^2 d + \beta)u \end{aligned} \quad (2.8)$$

La formulation faible se trouve comme suit :

Supposons que u est solution régulière de classe 2 du problème (2.6). Soit $v \in \mathcal{C}^1(\Omega)$ une fonction test, on multiplie l'équation de départ par v et on intègre sur Ω :

$$\int_{\Omega} -\nabla^2 u \cdot v + \int_{\Omega} \beta uv = \int_{\Omega} f v$$

on applique la formule de Green sur le terme d'ordre 2 :

$$\int_{\Omega} \nabla u \cdot \nabla v + \int_{\Omega} \beta uv = \int_{\Omega} f v + \int_{\partial\Omega} \frac{\partial u}{\partial n} \cdot v$$

En prenant en compte les conditions au bord, on obtient :

$$\int_{\Omega} \nabla u \cdot \nabla v + \int_{\Omega} \beta uv = \int_{\Omega} f v + v(1) \left[\frac{\partial u}{\partial x_1} \right]_{x_1=1} - v(0) \left[\frac{\partial u}{\partial x_1} \right]_{x_1=0} \quad (2.9)$$

La formulation faible est maintenant obtenue à partir de (2.9) en prenant des espaces de faible régularité, mais pour lesquels les intégrales existent. Ceci nous amène à prendre

$$u, v \in V = \{v \in H^1(\Omega) | v(x_1 = 0) = v(x_1 = 1) = 0\}$$

Par conséquent, d'après (2.9), la forme faible s'écrit :

$$\begin{cases} \text{chercher } u \in V, \text{ tq.} \\ a(u, v) = F(v) \quad \forall v \in V \end{cases} \quad (2.10)$$

où

$$\begin{aligned} a : V \times V &\longrightarrow \mathbb{R} \\ a(u, v) &= \int_{\Omega} \nabla u \cdot \nabla v + \int_{\Omega} \beta uv \\ F : V &\longrightarrow \mathbb{R} \\ F(v) &= \int_{\Omega} f v \end{aligned}$$

Nous pouvons ensuite regarder si la formulation faible obtenue admet une unique solution $u \in V$ grâce au théorème de Lax-Milgram.

Tout d'abord, V défini comme ci-dessus est un espace de Hilbert. Montrons ensuite, que la forme bilinéaire a est continue et coercive.

Pour la continuité, nous avons

$$\begin{aligned} |a(u, v)| &= \left| \int_{\Omega} \nabla u \cdot \nabla v + \int_{\Omega} \beta uv \right| \\ &\leq \int_{\Omega} |\nabla u \cdot \nabla v| + \int_{\Omega} |\beta| |uv| \\ &\leq \|\nabla u\|_{L^2} \|\nabla v\|_{L^2} + |\beta| \|u\|_{L^2} \|v\|_{L^2} \\ &\leq \|\nabla u\|_{L^2} \|\nabla v\|_{L^2} + C_p |\beta| \|\nabla u\|_{L^2} \|\nabla v\|_{L^2} \\ &\leq (1 + |\beta| C_p) \|\nabla u\|_{L^2} \|\nabla v\|_{L^2} \\ &\leq (1 + |\beta| C_p) \|u\|_{H^1} \|v\|_{H^1} \end{aligned}$$

Comme $(1 + |\beta| C_p) > 0$, nous pouvons en déduire la continuité pour tout β .

Regardons ce qu'il en est pour la coercivité. Si $\beta \geq 0$:

$$\begin{aligned} a(u, u) &= \int_{\Omega} (\nabla u)^2 + \int_{\Omega} \beta u^2 \\ &= \|\nabla u\|_{L^2}^2 + \int_{\Omega} \beta u^2 \\ &\geq \|\nabla u\|_{L^2}^2 = \|u\|_{H^1}^2 \end{aligned}$$

et si $\beta < 0$:

$$\begin{aligned} a(u, u) &= \int_{\Omega} (\nabla u)^2 + \int_{\Omega} \beta u^2 \\ &= \|\nabla u\|_{L^2}^2 + \int_{\Omega} \beta u^2 \\ &\geq \|\nabla u\|_{L^2}^2 + \beta C_p \|\nabla u\|_{L^2}^2 \\ &= (1 + \beta C_p) \|\nabla u\|_{L^2}^2 \\ &= (1 + \beta C_p) \|u\|_{H^1}^2 \end{aligned}$$

Pour que la condition de coercivité soit vérifiée, il faut que $(1 + \beta C_p) > 0$ et donc que $\beta > -1/C_p$. Nous ne connaissons pas la valeur exacte de la constante de Poincaré sur $\Omega = [0, 1]^3$, mais nous savons qu'elle est positive. Nous pouvons donc en déduire que pour tout $\beta \geq 0$, la coercivité est vérifiée. Mais nous ne pouvons pas nous prononcer quant à la coercivité de $\beta = -10$. Il faudra donc attendre les résultats des test pour faire une comparaison du comportement des normes entre les différents β .

Montrons finalement la continuité de la forme linéaire F

$$\begin{aligned} |F(v)| &= \left| \int_{\Omega} f v \right| \\ &\leq \|f\|_{L^2} \|v\|_{L^2} \\ &\leq C_p \|f\|_{L^2} \|\nabla v\|_{L^2} \\ &= C_p \|f\|_{L^2} \|v\|_V \end{aligned}$$

Nous avons donc vérifié l'existence et l'unicité de la solution u .

Par la méthode des éléments finis implémentée en C++, le programme nous calcule une approximation u_h que nous allons analyser dans le prochain chapitre.

Comme nous l'avons vu à la section 1.2, la résolution du problème nécessite un maillage. En 3 dimensions, nous utilisons une décomposition tétraédrique du domaine. Le domaine de nos deux problèmes étant $\Omega = [0, 1]^3$, nous générons un maillage du cube à l'aide du programme *gmsh*. Nous en donnons deux exemples dans les figures 2.1 et 2.2 qui correspondent à une décomposition des droites en 3 ($h = \frac{1}{4}$), respectivement 15 ($h = \frac{1}{16}$) parties.

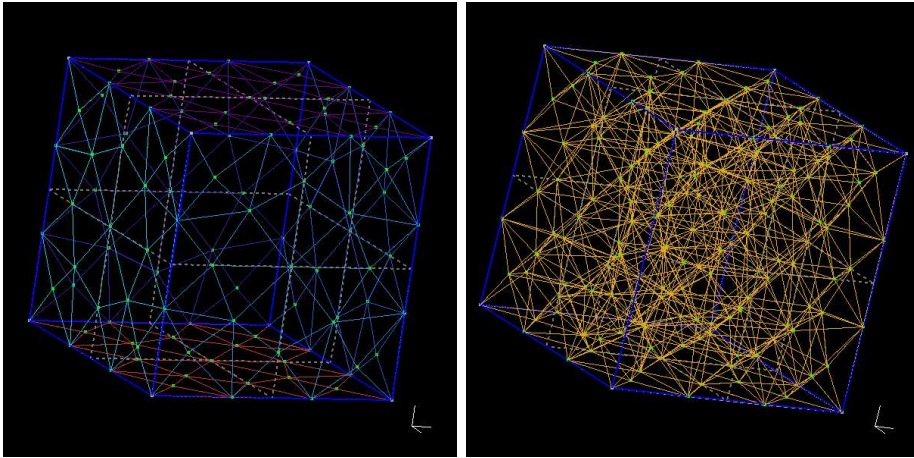


FIG. 2.1 – Maillage du cube pour $h = \frac{1}{4}$. A droite, maillage du volume (3D) et à gauche, maillage des faces du cube (2D).

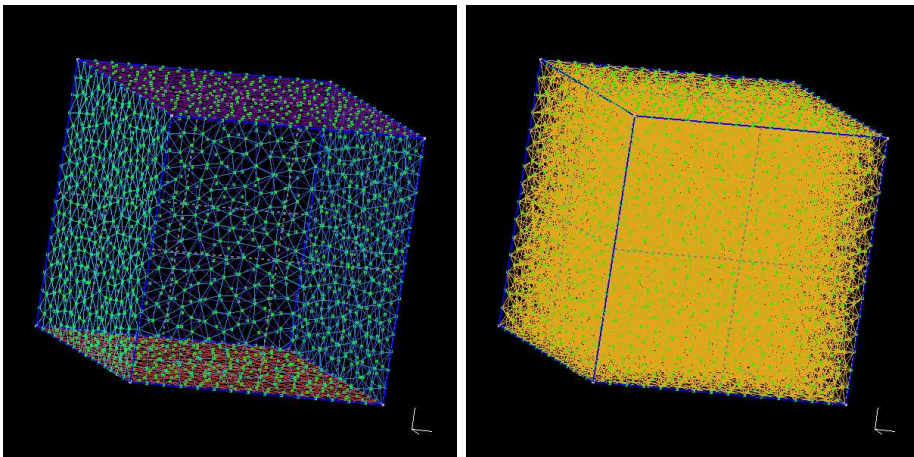
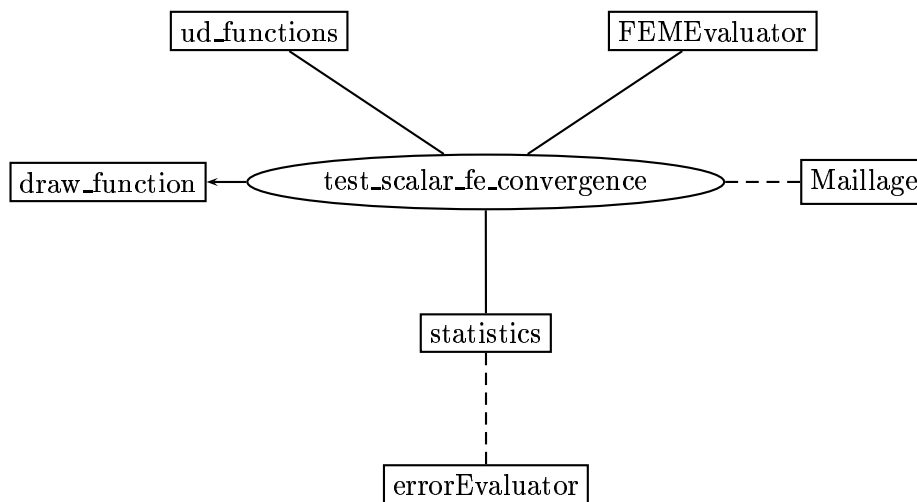


FIG. 2.2 – Maillage du cube pour $h = \frac{1}{16}$. A droite, maillage du volume (3D) et à gauche, maillage des faces du cube (2D).

Chapitre 3

Programmation

De nombreux programmes ont été implémentés par les développeurs de LifeV ; ils forment une architecture très complexe. Nous n'avons utilisé ou implémenté qu'un petit nombre de programmes, dont la structure (très simplifiée) peut être représentée graphiquement comme suit :



3.1 Description des programmes

Nous allons brièvement décrire chaque programme, mais seulement ce que nous avons utilisé. En effet, tous les programmes que nous avons cité

ci-dessus servent à résoudre trois problèmes : $x2y2z2$, $sincosd$ et $tx2y2z2$ qui est une généralisation de $x2y2z2$ avec composante temporelle. Nous n'avons pas traité ce dernier problème.

Tout d'abord, nous avons *ud_ functions*, où sont implémentées deux classes : $x2y2z2$ et $sincosd$. Dans ces classes, nous pouvons trouver l'évaluation pour tout vecteur x de chacune de ces fonctions, ainsi que le calcul du gradient, de la matrice hessienne et du laplacien correspondant à chacune des fonctions.

Ensuite, nous avons implémenté la classe *Statistics*. Cette classe nous permet de calculer la moyenne, la variance, la covariance et le carré moyen de l'erreur des valeurs stockées dans un vecteur de taille $n \times 2$. En particulier, le carré moyen de l'erreur nous donne la régression linéaire d'une fonction $f(h)$ (les deux colonnes de notre vecteur seront alors h et $f(h)$).

Dans le programme *FEMEvaluator*, nous avons implémenté la forme bilinéaire a et le second membre F pour chacun de nos problèmes. Il nous permet ainsi, à l'aide d'un solveur linéaire, de trouver l'approximation u_h de u et de calculer la norme d'un vecteur quelconque dans les espaces L^1, L^2, L^∞, H^1 , ainsi que la norme de l'énergie. Il calcule également $\frac{\|e\|_{H^1}}{|e|_{H^2}}$ et $\frac{\|e\|_{L^2}}{|e|_{H^2}}$

Finalement, le fichier *test_scalar_fe_convergence* est le main de nos programmes. Grâce aux classes *PrintError* et *LatexPrintError*, il écrit les valeurs calculées par *FEMEvaluator*, dans des fichiers *.txt* et *.tex*, dont nous aurons besoin par la suite. *StatisticsError* contrôle le test et informe l'utilisateur lorsque les valeurs obtenues ne sont pas celles attendues (par la théorie).

Afin de mieux pouvoir visualiser les résultats, nous avons implémenté une fonction Matlab, nommée *draw_function*. Cette fonction lit les valeurs numériques dans les fichiers *.txt* écrits par *test_scalar_fe_convergence* et dessine, en mode Log-Log, $\frac{\|e\|_{H^1}}{|e|_{H^2}}$ (respectivement $\frac{\|e\|_{L^2}}{|e|_{H^2}}$) en fonction de h (le pas de maillage), ainsi que sa régression linéaire. De plus, on représente sur le même graphique les droites de pente 1 (fonction h) et de pente 2 (fonction h^2) pour favoriser la comparaison avec la théorie.

Nous avons aussi écrit deux autres programmes au début de notre projet, mais ils ne sont pas directement liés au test final. Le premier est *Maillage* ; il s'agit d'une classe qui génère un maillage en une dimension. L'autre est *ErrorEvaluator* ; cette classe permet d'évaluer une fonction dans une certaine valeur et écrit ces valeurs dans un fichier *.txt*, de manière à pouvoir les dessiner avec Matlab. Elle fait en fait la même chose que *test_scalar_fe_convergence*, de manière beaucoup plus facile. Elle nous a, entre autre, permis de vérifier le fonctionnement de la classe *Statistics* et le code Matlab. Nous avons d'ailleurs constaté un problème avec la méthode *stddev* de *Statistics*, qui calcule le carré moyen de l'erreur ; en effet, la pente calculée par la méthode

slope devrait être la même que la valeur que retourne la méthode *stddev*, ce qui est contraire à nos observations.

3.2 Design pattern

Nous avons eu l'occasion dans le cadre de ce projet, de nous familiariser avec les *design pattern*.

Il s'agit d'un motif de conception dont le rôle est d'implémenter un mécanisme générique répondant à une problématique récurrente sur un projet. Il nous permet donc de réutiliser une solution qui a déjà fait ses preuves autant qu'on le désire et instaure une possibilité de déléguer les tâches.

Le pattern *Observer* quant à lui définit une dépendance entre objets de telle manière que lorsque l'un des objets change d'états, tous les autres objets dépendants sont avertis et mis à jour. C'est donc grâce à *Observer* qu'il nous a été possible de créer la structure présentée ci-dessus.

Chapitre 4

Résultats

Nous cherchons à vérifier si nous obtenons des résultats conformes à ce que dit la théorie et en particulier à vérifier les estimations suivantes (cf. corollaire 1.5) :

$$\|u - u_h\|_{L^2(\Omega)} \leq Ch^2 |u|_{H^2(\Omega)} \quad (4.1)$$

$$\|u - u_h\|_{H^1(\Omega)} \leq Ch |u|_{H^2(\Omega)} \quad (4.2)$$

4.1 Le problème $x^2y^2z^2$

Nous présentons les résultats indépendamment pour chaque β . Les tableaux 4.1-4.6) donnent les valeurs pour $\|u - u_h\|$, selon chacune des cinq normes étudiées (ie. L^1 , L^2 , L^∞ , H^1 et la norme de l'énergie, notée a), et pour différentes valeurs de h .

Nous avons commencé par travailler dans \mathbb{P}^1 et donc considéré l'espace des éléments finis de Lagrange de degré 1. Dans les tableaux 4.1-4.3, on peut aisément constater que les inéquations (4.1) et (4.2), données par la théorie sont satisfaites. Par ailleurs, on peut remarquer que le système se comporte de manière semblable pour $\beta = -10$ que pour $\beta \geq 0$. Ceci implique soit que β satisfait la condition imposée par la coercivité ($\beta > -1/C_p$), soit alors que même sans satisfaire le cadre théorique, on obtient un test qui fonctionne. On pourrait calculer numériquement la constante C_p pour vérifier cette hypothèse.

Ainsi, les tests de convergence semblent être efficaces ; mais nous ne pouvons pas nous baser sur un seul problème, d'autant plus que ce problème admet une solution exacte polynômiale d'ordre 2. En effet, si nous choisissons maintenant l'espace des éléments finis de Lagrange de degré 2, nous remarquons que toutes les valeurs calculées sont approximativement égales à 0 (cf. tableaux 4.4-4.6).

En réalité, nous nous attendions à ces résultats. En effet, comme \mathbb{P}^2 est l'espace qui engendre tous les polynôme de degré inférieur ou égal à 2, toute

approximation d'ordre 2 donnera la solution exacte.

Les figures 4.1-4.6 représentent, en mode Log-Log, $\frac{\|u-u_h\|_{H^1(\Omega)}}{\|u\|_{H^2(\Omega)}}$, respectivement $\frac{\|u-u_h\|_{L^2(\Omega)}}{\|u\|_{H^2(\Omega)}}$ en fonction de h (courbe verte). A l'aide de leur régression linéaire (courbe magenta) et des courbes de pente 1 (courbe rouge) et de pente 2 (courbe bleue), nous pouvons visualiser graphiquement si les valeurs théoriques, données par les inéquations (4.1) et (4.2), sont satisfaites. En effet, en mode Log-Log, une courbe d'ordre h^r doit avoir une pente de r .

Nous constatons alors que dans les graphiques 4.1-4.3, ces approximations sont satisfaites. Mais, dans les graphiques 4.4-4.6, on trouve des pentes négatives. Or, nous pouvons expliquer ceci, par le fait que $\|u-u_h\|_{H^1}$ et $\|u-u_h\|_{L^2}$ sont quasi-nulles (donc les pentes ne sont pas représentatives).

Pour remédier au problème de trouver la solution exacte lors de l'approximation, nous prenons un problème non polynômial, décrit à la section (2.2) et dont les résultats sont présentés à la section suivante.

h	L^1	L^2	L^∞	a	H^1	L^2 normalisé	H^1 normalisé
1/4	0.00985686	0.0097478	0.0199327	0.0596833	0.0738465	0.00216923	0.0164334
1/8	0.00446075	0.00429243	0.00884621	0.0327973	0.0305082	0.000972079	0.00690901
1/16	0.000615836	0.000686919	0.00202244	0.00973592	0.0105681	0.000178321	0.00274345
1/32	0.000242295	0.000218746	0.000515849	0.00481371	0.00524834	4.66662e-05	0.00111965

TAB. 4.1 – Résultats pour le problème $x2y2z2$ en \mathbb{P}^1 avec $\beta = -10$

h	L^1	L^2	L^∞	a	H^1	L^2 normalisé	H^1 normalisé
1/4	0.00340224	0.00352435	0.0119551	0.0467511	0.05236	0.000784292	0.0116519
1/8	0.00152843	0.00149459	0.00806664	0.028988	0.0225988	0.00033847	0.00511781
1/16	0.000151867	0.000177208	0.00142149	0.00932874	0.00971016	4.60024e-05	0.00252072
1/32	6.07823e-05	5.72662e-05	0.000381414	0.00475779	0.00511495	1.22169e-05	0.0010912

TAB. 4.2 – Résultats pour le problème $x2y2z2$ en \mathbb{P}^1 avec $\beta = 0$

h	L^1	L^2	L^∞	a	H^1	L^2 normalisé	H^1 normalisé
1/4	0.0227809	0.0214553	0.0408153	0.560576	0.16067	0.00477455	0.0357546
1/8	0.0066635	0.00585542	0.00851222	0.155139	0.0509709	0.00132604	0.0115431
1/16	0.0013966	0.00135029	0.00222204	0.0415229	0.0177574	0.000350531	0.00460975
1/32	0.000524074	0.000411979	0.000553825	0.0112001	0.00642657	8.78894e-05	0.00137101

TAB. 4.3 – Résultats pour le problème $x2y2z2$ en \mathbb{P}^1 avec $\beta = 1000$

h	L^1	L^2	L^∞	a	H^1	L^2 normalisé	H^1 normalisé
1/4	4.86818e-12	6.19031e-12	2.8966e-11	9.10284e-11	1.20085e-10	1.40534e-12	2.72621e-11
1/8	7.99123e-12	8.01789e-12	3.21767e-11	1.20275e-10	1.401e-10	1.6539e-12	2.88993e-11
1/16	2.00339e-11	2.02519e-11	1.2798e-10	2.97605e-10	3.58507e-10	4.32851e-12	7.66249e-11

TAB. 4.4 – Résultats pour le problème $x2y2z2$ en \mathbb{P}^2 avec $\beta = -10$

h	L^1	L^2	L^∞	a	H^1	L^2 normalisé	H^1 normalisé
1/4	1.24227e-11	1.69229e-11	8.62344e-11	2.24584e-10	2.89834e-10	3.84188e-12	6.5799e-11
1/8	8.85634e-12	8.99757e-12	3.84937e-11	1.39427e-10	1.60412e-10	1.85598e-12	3.30891e-11
1/16	3.28254e-11	3.71116e-11	1.41604e-10	3.8929e-10	4.72922e-10	7.93199e-12	1.01079e-10

TAB. 4.5 – Résultats pour le problème $x2y2z2$ en \mathbb{P}^2 avec $\beta = 0$

h	L^1	L^2	L^∞	a	H^1	L^2 normalisé	H^1 normalisé
1/4	2.6724e-12	3.41209e-12	1.89386e-11	1.2048e-10	8.73879e-11	7.74623e-13	1.98391e-11
1/8	1.73908e-12	1.82886e-12	1.59442e-11	7.19682e-11	6.38328e-11	3.77249e-13	1.31672e-11
1/16	2.12449e-11	2.25823e-11	4.53734e-10	1.29057e-09	1.32193e-09	4.8266e-12	2.82541e-10

TAB. 4.6 – Résultats pour le problème $x2y2z2$ en \mathbb{P}^2 avec $\beta = 1000$

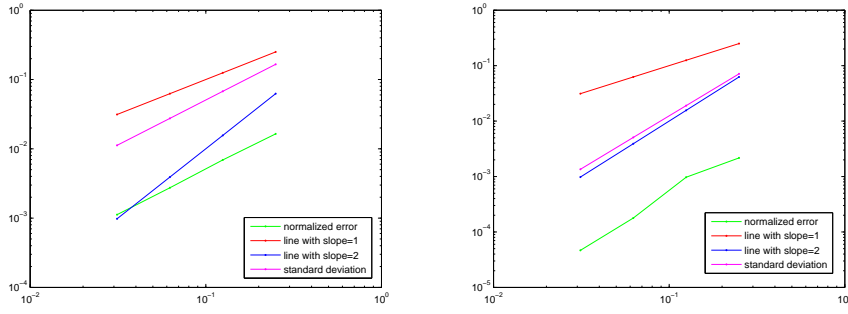


FIG. 4.1 – Graphique de la fonction $x^2y^2z^2$ pour $\beta = -10$ en \mathbb{P}^1 . A gauche, on a $\frac{\|u-u_h\|_{H^1(\Omega)}}{|u|_{H^2(\Omega)}}$ en vert, et à droite, $\frac{\|u-u_h\|_{L^2(\Omega)}}{|u|_{H^2(\Omega)}}$.

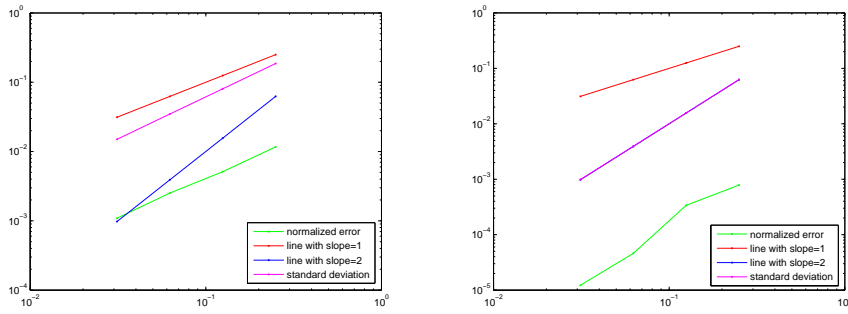


FIG. 4.2 – Graphique de la fonction $x^2y^2z^2$ pour $\beta = 0$ en \mathbb{P}^1 . A gauche, on a $\frac{\|u-u_h\|_{H^1(\Omega)}}{|u|_{H^2(\Omega)}}$ en vert, et à droite, $\frac{\|u-u_h\|_{L^2(\Omega)}}{|u|_{H^2(\Omega)}}$.

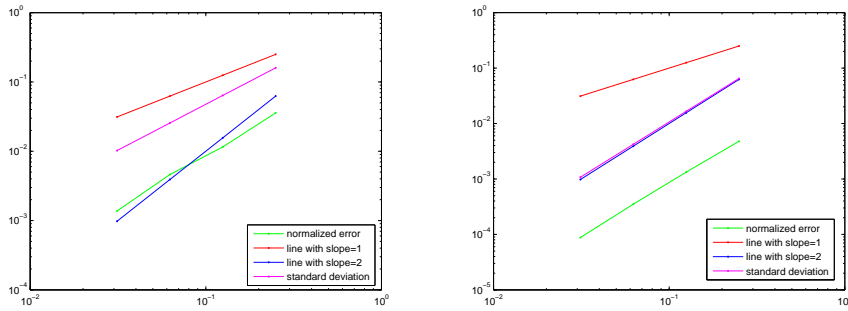


FIG. 4.3 – Graphique de la fonction $x^2y^2z^2$ pour $\beta = 1000$ en \mathbb{P}^1 . A gauche, on a $\frac{\|u-u_h\|_{H^1(\Omega)}}{|u|_{H^2(\Omega)}}$ en vert, et à droite, $\frac{\|u-u_h\|_{L^2(\Omega)}}{|u|_{H^2(\Omega)}}$.

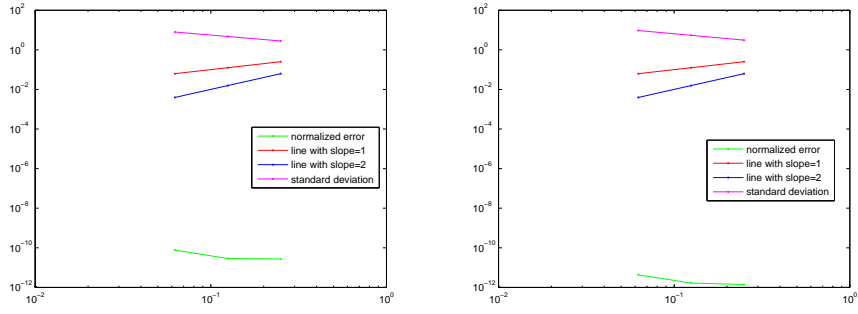


FIG. 4.4 – Graphique de la fonction $x^2y^2z^2$ pour $\beta = -10$ en \mathbb{P}^2 . A gauche, on a $\frac{\|u-u_h\|_{H^1(\Omega)}}{|u|_{H^2(\Omega)}}$ en vert, et à droite, $\frac{\|u-u_h\|_{L^2(\Omega)}}{|u|_{H^2(\Omega)}}$.

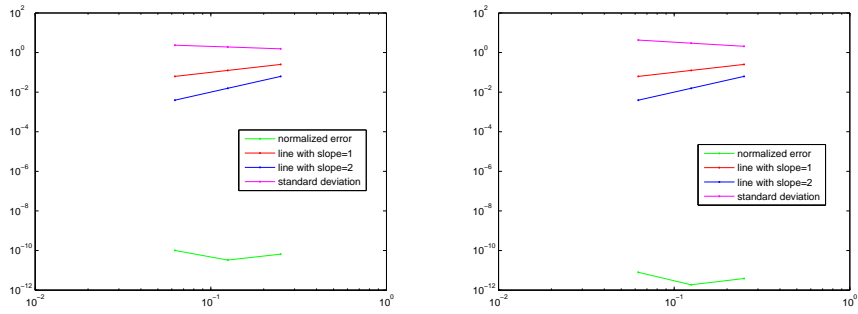


FIG. 4.5 – Graphique de la fonction $x^2y^2z^2$ pour $\beta = 0$ en \mathbb{P}^2 . A gauche, on a $\frac{\|u-u_h\|_{H^1(\Omega)}}{|u|_{H^2(\Omega)}}$ en vert, et à droite, $\frac{\|u-u_h\|_{L^2(\Omega)}}{|u|_{H^2(\Omega)}}$.

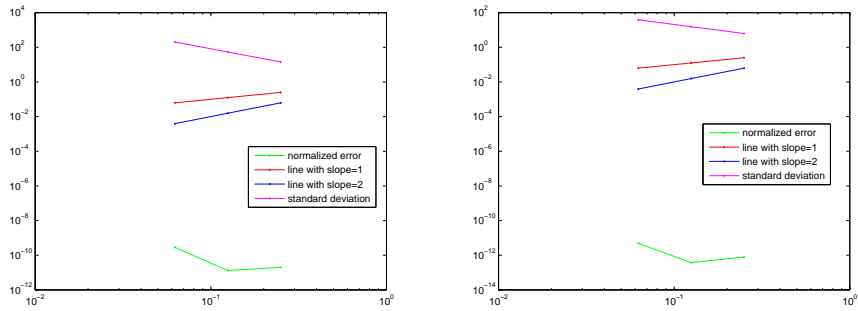


FIG. 4.6 – Graphique de la fonction $x^2y^2z^2$ pour $\beta = 1000$ en \mathbb{P}^2 . A gauche, on a $\frac{\|u-u_h\|_{H^1(\Omega)}}{|u|_{H^2(\Omega)}}$ en vert, et à droite, $\frac{\|u-u_h\|_{L^2(\Omega)}}{|u|_{H^2(\Omega)}}$.

4.2 Le problème *sincosd*

Comme pour le problème précédent, nous avons rempli un tableaux par β pour \mathbb{P}^1 et \mathbb{P}^2 respectivement (cf. tableaux 4.7-4.12). Par contre, nous avons constaté un problème lors de l'exécution et avons dû changer de solveur pour pouvoir résoudre le problème. On en déduit que la matrice A du système est mal-conditionnée. Pour cette même raison, nous n'avons de résultats que pour trois valeurs de h . Ceci suffit néanmoins pour l'analyse des résultats.

A nouveau, nous pouvons constater que pour la valeur $\beta = -10$, les résultats sont comparables à ceux obtenus pour $\beta \geq 0$. En ce qui concerne la convergence, les résultats ne sont pas aussi précis que pour le problème polynômial, cependant l'approximation est acceptable.

Les graphiques en \mathbb{P}^1 ne sont pas très satisfaisant (la régression linéaire pour L^2 est plus proche de la courbe rouge que de la courbe bleue), mais on note une nette amélioration lorsqu'on passe aux éléments finis d'ordre 2. Ceci confirme les résultats théoriques.

h	L^1	L^2	L^∞	a	H^1	L^2 normalisé	H^1 normalisé
1/4	0.0432694	0.061733	0.188688	0.554122	0.735668	0.00458139	0.0545961
1/8	0.0244319	0.0343198	0.265554	0.385516	0.446574	0.00234177	0.0304715
1/16	0.00901384	0.0116523	0.165285	0.218208	0.27563	0.000824046	0.0194925

TAB. 4.7 – Résultats pour le problème *sincosd* en \mathbb{P}^1 avec $\beta = -10$

h	L^1	L^2	L^∞	a	H^1	L^2 normalisé	H^1 normalisé
1/4	0.0439499	0.0603567	0.33321	0.544086	0.699159	0.00447925	0.0518867
1/8	0.0234069	0.0325698	0.256003	0.381239	0.431556	0.00222236	0.0294467
1/16	0.00666531	0.0105296	0.162563	0.217079	0.272273	0.000744652	0.019255

TAB. 4.8 – Résultats pour le problème *sincosd* en \mathbb{P}^1 avec $\beta = 0$

h	L^1	L^2	L^∞	a	H^1	L^2 normalisé	H^1 normalisé
1/4	0.0439499	0.0408081	0.0958751	1.02508	0.323144	0.00302849	0.0239815
1/8	0.0234069	0.015704	0.0847735	0.393388	0.19526	0.00107155	0.0133233
1/16	0.00666531	0.00515064	0.0812636	0.188688	0.179918	0.000364251	0.0127238

TAB. 4.9 – Résultats pour le problème *sincosd* en \mathbb{P}^1 avec $\beta = 1000$

h	L^1	L^2	L^∞	a	H^1	L^2 normalisé	H^1 normalisé
1/4	0.0160311	0.0204868	0.206763	0.258882	0.0204868	0.00150143	0.00150143
1/8	0.00731704	0.008658	0.126291	0.16531	0.008658	0.000646814	0.000646814
1/16	0.00113352	0.00184341	0.0787837	0.0815771	0.00184341	0.000158514	0.000158514

TAB. 4.10 – Résultats pour le problème *sincosd* en \mathbb{P}^2 avec $\beta = -10$

h	L^1	L^2	L^∞	a	H^1	L^2 normalisé	H^1 normalisé
1/4	0.0117662	0.0174699	0.20163	0.255886	0.0174699	0.00128033	0.00128033
1/8	0.00468959	0.00698836	0.124299	0.164306	0.00698836	0.000522081	0.000522081
1/16	0.000827667	0.00162381	0.0783923	0.0814302	0.00162381	0.000139631	0.000139631

TAB. 4.11 – Résultats pour le problème *sincosd* en \mathbb{P}^2 avec $\beta = 0$

h	L^1	L^2	L^∞	a	H^1	L^2 normalisé	H^1 normalisé
1/4	0.00252273	0.00568206	0.104238	0.194053	0.00568206	0.000416426	0.000416426
1/8	0.000625081	0.00277598	0.0855425	0.135288	0.00277598	0.000207386	0.000207386
1/16	0.000108461	0.0008871	0.0627044	0.075557	0.0008871	7.62816e-05	7.62816e-05

TAB. 4.12 – Résultats pour le problème *sincosd* en \mathbb{P}^2 avec $\beta = 1000$

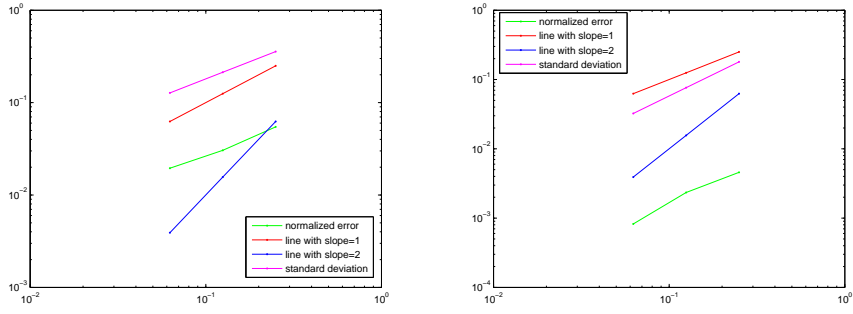


FIG. 4.7 – Graphique de la fonction sincosd pour $\beta = -10$ en \mathbb{P}^1 . A gauche, on a $\frac{\|u - u_h\|_{H^1(\Omega)}}{|u|_{H^2(\Omega)}}$ en vert, et à droite, $\frac{\|u - u_h\|_{L^2(\Omega)}}{|u|_{H^2(\Omega)}}$.

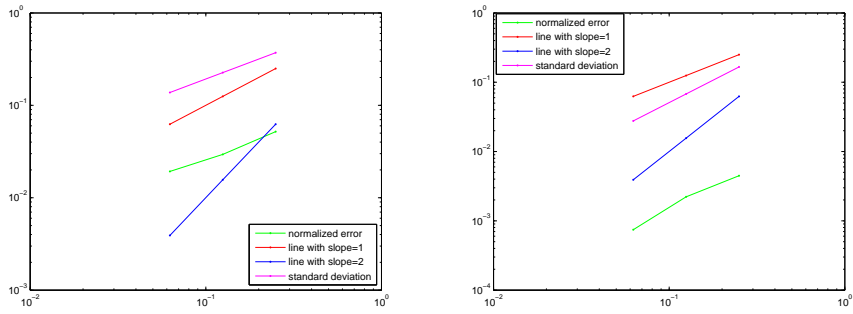


FIG. 4.8 – Graphique de la fonction sincosd pour $\beta = 0$ en \mathbb{P}^1 . A gauche, on a $\frac{\|u - u_h\|_{H^1(\Omega)}}{|u|_{H^2(\Omega)}}$ en vert, et à droite, $\frac{\|u - u_h\|_{L^2(\Omega)}}{|u|_{H^2(\Omega)}}$.

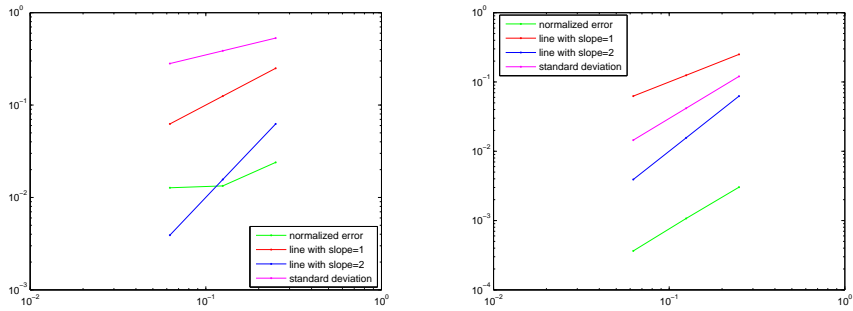


FIG. 4.9 – Graphique de la fonction sincosd pour $\beta = 1000$ en \mathbb{P}^1 . A gauche, on a $\frac{\|u - u_h\|_{H^1(\Omega)}}{|u|_{H^2(\Omega)}}$ en vert, et à droite, $\frac{\|u - u_h\|_{L^2(\Omega)}}{|u|_{H^2(\Omega)}}$.

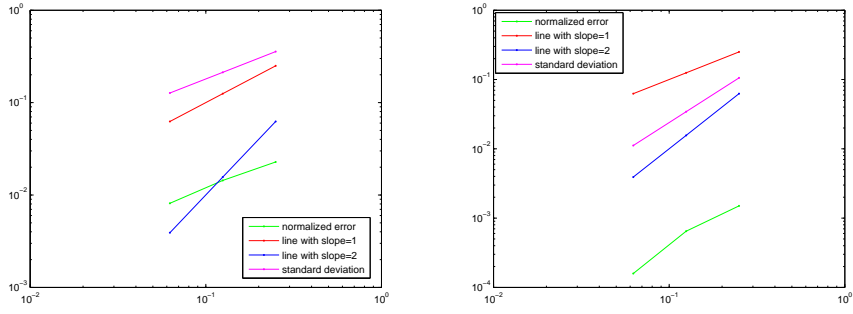


FIG. 4.10 – Graphique de la fonction *sincosd* pour $\beta = -10$ en \mathbb{P}^2 . A gauche, on a $\frac{\|u-u_h\|_{H^1(\Omega)}}{\|u\|_{H^2(\Omega)}}$ en vert, et à droite, $\frac{\|u-u_h\|_{L^2(\Omega)}}{\|u\|_{H^2(\Omega)}}$.

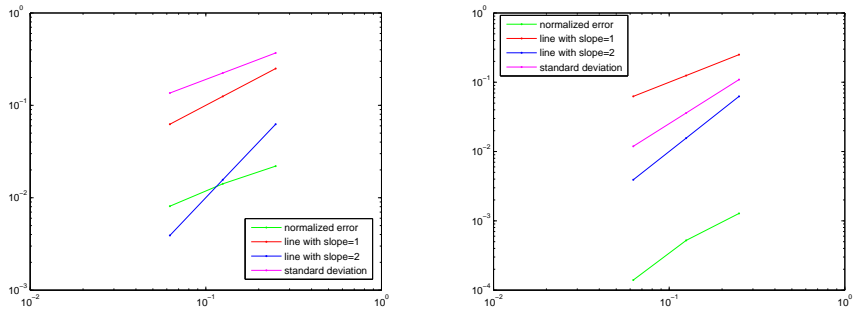


FIG. 4.11 – Graphique de la fonction *sincosd* pour $\beta = 0$ en \mathbb{P}^2 . A gauche, on a $\frac{\|u-u_h\|_{H^1(\Omega)}}{\|u\|_{H^2(\Omega)}}$ en vert, et à droite, $\frac{\|u-u_h\|_{L^2(\Omega)}}{\|u\|_{H^2(\Omega)}}$.

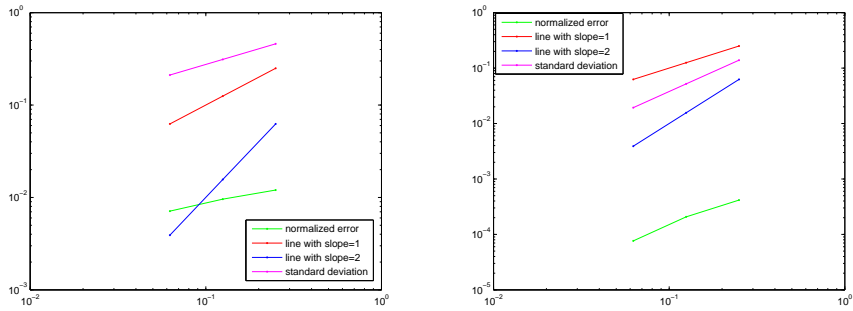


FIG. 4.12 – Graphique de la fonction *sincosd* pour $\beta = 1000$ en \mathbb{P}^2 . A gauche, on a $\frac{\|u-u_h\|_{H^1(\Omega)}}{\|u\|_{H^2(\Omega)}}$ en vert, et à droite, $\frac{\|u-u_h\|_{L^2(\Omega)}}{\|u\|_{H^2(\Omega)}}$.

Chapitre 5

Conclusion

Ce projet nous a permis de généraliser la théorie vue au cours à la théorie qui est effectivement appliquée en pratique. Nous avons également eu l'occasion de perfectionner nos connaissances en programmation et avons eu l'opportunité de travailler au sein d'un véritable groupe de recherche.

Nous constatons que les résultats obtenus sont très satisfaisant, mais nous avons été confrontés à des problèmes dus à des erreurs dans le code. Nous avons ainsi pu contribuer de manière constructive au développement du projet LifeV.

Le travail effectué n'est de loin pas abouti, en effet plusieurs obstacles sont survenus à la fin et nous n'avons malheureusement pas eu assez de temps pour analyser et résoudre nous-mêmes ces derniers problèmes. Mais les développeurs de LifeV seront ravis de faire ce travail.

Par exemple, nous avons constaté des problèmes de solveur lorsque nous nous sommes occupés du problème *sincosd*. Il aurait été intéressant de faire une analyse plus détaillée et quelques tests supplémentaires pour voir la fiabilité des solveurs. Il y a alors plusieurs possibilités pour choisir le solveur : soit on choisit un solveur robuste qui fonctionne pour tous les cas testés, ou alors on change de solveur pour les différentes valeurs de β ou de h , afin d'avoir de meilleures convergences.

Il est à déplorer que nous ayons manqué de temps, car nous aurions voulu rendre les tests entièrement automatiques, de manière à ce qu'en une nuit, il vérifie tous les cas.

Pour améliorer les tests de convergence, nous aurions pu améliorer nos programmes de manière à ce qu'il teste également d'autres espaces. En effet, nous avons restreints notre analyse aux espaces de polynômes linéaires et quadratiques, \mathbb{P}^1 et \mathbb{P}^2 , mais il existe également d'autres espaces intéressants pour l'analyse des éléments finis, comme par exemple \mathbb{Q}^k .

Un autre développement pourrait se faire à des problèmes plus complexes, qui prennent en compte une variation dans le temps (problèmes paraboliques ou hyperboliques), comme on le rencontre par exemple dans un

problème classique de physique : l'équation de chaleur (ie., pour le cas monodimensionnel, l'évolution de la température au point x et à l'instant t d'une barre métallique).

Bibliographie

- [1] <http://cmcsforge.epfl.ch/>.
- [2] <http://cvsbook.redbean.com/cvsbook.html>.
- [3] <http://www.lifev.org/>.
- [4] http://www.pstmartin.freesurf.fr/gestioncontenu/perso/download/pdf/Design_patterns.pdf.
- [5] Marylène Micheloud and Medard Rieder. *Programmation orientée objet en C++*. PPUR, 2003.
- [6] Alfio Quarteroni and Marco Discacciati. *Cours d'éléments d'approximations des équations aux dérivées partielles*. EPFL, 2004-2005.
- [7] Alfio Quarteroni, Riccardo Sacco, and Fausto Saleri. *Méthodes numériques pour le calcul scientifique*. Springer, 2000.