ANN201 TP1

GIOVANNINI Mathéo BOULOGNE Pierre

1 Introduction

Dans ce TP, nous allons résoudre à l'aide de Matlab un problème de Poisson avec condition aux limites de Neumann:
Trouver $u \in H^1(\Omega)$ telle que

$$\begin{cases} u - \Delta u = f \text{ dans } \Omega \\ \frac{\partial u}{\partial n} = 0 \text{ sur } \partial\Omega \end{cases}$$

Nous allons d'abord écrire la formulation variationnelle du problème afin de vérifier s'il est bien posé et de le manipuler par la suite.
Soit u vérifiant : $\forall v \in H^1(\Omega)$, $\int_\Omega (uv - \Delta uv) d\Omega = \int_\Omega fv d\Omega$

En appliquant la formule de Green:

$$\int_\Omega uv d\Omega + \int_\Omega \nabla u \cdot \nabla v d\Omega - \int_\Omega \frac{\partial u}{\partial n} v d\Gamma = \int_\Omega fv d\Omega$$

Or $\frac{\partial u}{\partial n} = 0$ sur Ω donc on obient la formulation variationnelle :

$$\int_\Omega uv d\Omega + \int_\Omega \nabla u \cdot \nabla v d\Omega = \int_\Omega fv d\Omega \tag{FV}$$

On pose alors

$$a(u,v) = \int_\Omega uv d\Omega + \int_\Omega \nabla u \cdot \nabla v d\Omega \; et \; l(v) = \int_\Omega fv d\Omega$$

Vérifions que le problème est bien posé:
$(H^1(\Omega), ||.||)$ est un espace de Hilbert
a est bilinéaire, continue car :

$$\begin{aligned} |a(u,v)| &\leqslant \int_\Omega |uv| d\Omega + \int_\Omega |\nabla u \cdot \nabla v| d\Omega \\ &\leqslant ||u||_{L^2(\Omega)} \cdot ||v||_{L^2(\Omega)} + ||\nabla u||_{L^2(\Omega)} \cdot ||\nabla v||_{L^2(\Omega)} \\ &\leqslant 2||u||_{H^1(\Omega)} \cdot ||v||_{H^1(\Omega)} \end{aligned}$$

et coercive car : $a(u,u) = ||u||^2_{H^1(\Omega)} \geqslant 1 \cdot ||u||^2_{H^1(\Omega)}$
D'autre part, ℓ est linéaire et est continue car :

$$\left| \int_\Omega fv d\Omega \right| \leqslant ||f||_{L^2(\Omega)} \cdot ||v||_{L^2(\Omega)} \leqslant ||f||_{L^2(\Omega)} \cdot ||v||_{H^1(\Omega)}$$

D'après le théorème de Lax-Milgram, le problème est bien posé.

Afin d'étudier plus facilement le problème avec Matlab, on va chercher à le linéariser. Pour cela, on pose T_h une triangulation du domaine Ω et on approche $H^1(\Omega)$ par des éléments finis P^1 associés à la triangulation. On note V_h l'approximation de $H^1(\Omega)$ obtenue.

La formulation variationnelle devient:

$$\int_\Omega u_h v_h d\Omega + \int_\Omega \nabla u_h \cdot \nabla v_h d\Omega = \int_\Omega f v_h d\Omega \qquad (FV_h)$$

On linéarise alors le système.
Comme $V_h = Vect((w_i)_{i\in[\![1,N]\!]})$ et que $w_i(S_j) = \delta_{ij}$, la solution approchée u_h vérifie : $\forall (x,y) \in \bar{\Omega},\ u_h(x,y) = \sum_{I=1}^N u_h(S_I) w_I(x,y)$
$(w_i)_{i\in[\![1,N]\!]}$ étant une base de V_h on remarque que u_h est solution de (FV_h) si et seulement si u_h vérifie :

$$\forall j \in [\![1,N]\!],\ \int_\Omega u_h w_j d\Omega + \int_\Omega \nabla u_h \cdot \nabla w_j d\Omega = \int_\Omega f w_j d\Omega$$

On a alors :

$$(FV_h) \Leftrightarrow \forall i \in [\![1,N]\!],\ \sum_{j=1}^N \left(u_h(S_j) \int_\Omega w_i w_j d\Omega \right) + \sum_{j=1}^N \left(u_h(S_j) \int_\Omega \nabla w_i \cdot \nabla w_j d\Omega \right) = \int_\Omega f w_i d\Omega$$

Finalement, (FV_h) est équivalent au système matriciel : $(\mathbb{M} + \mathbb{K})\vec{U} = \vec{L}$.
Avec :

$$\begin{aligned} \forall (i,j) \in [\![1,N]\!]^2,\ \mathbb{M}_{i,j} &= \int_\Omega w_i w_j d\Omega \\ \mathbb{K}_{i,j} &= \int_\Omega \nabla w_i \cdot \nabla w_j d\Omega \\ \vec{U}_i &= u_h(S_i) \\ \vec{L}_i &= \int_\Omega f w_j d\Omega = \ell(w_i) \end{aligned}$$

a étant coercive, $\mathbb{M}+\mathbb{K}$ est définie positive, donc inversible, de plus, ces matrices sont creuses, c'est une propriété inhérente au maillage choisi. Pour tout j dans $[\![1,N]\!]$, w_j est nulle sur les triangles qui n'ont pas le point S_j comme sommet. Il suffit donc que S_j et S_i soient "assez éloignés" pour que les supports de w_i et w_j soient disjoints et donc que $\int_\Omega w_i w_j$ soit nul, ce qui est le cas pour la plupart des couples de sommets. Les matrices $\mathbb{M}$ et $\mathbb{K}$ sont donc principalement constituées de 0 en général. Cette dernière propriété est très utile pour traiter ces matrices numériquement.

On crée les maillages avec le logiciel Gmsh, en voici un exemple :

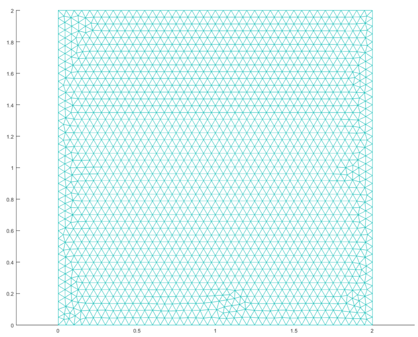

Figure 1: Maillage

En ouvrant le fichier avec Matlab, on obtient une structure de données caractérisant le maillage :

Nbpt est un entier donnant le nombre de points dans le maillage.
Nbtri est un entier donnant le nombre de triangles.
Coorneu est une matrice donnant les coordonées des sommets.
Refneu est un vecteur donnant les sommets sur le bord.
Numtri est une matrice donnant les sommets associés à chaque triangle.
Reftri est un vectuer donnant la référence des triangles.
Nbaretes est un entier donnant le nombre d'arêtes (sur le bord).
Numaretes est un vecteur donnant les sommetes liées aux arêtes.
Refaretes est un vecteur donnant la référence des arêtes.

On calcule à présent les matrices élémentaires $\mathbb{M}^{élém}$ et $\mathbb{K}^{élém}$ sur les différents triangles à partir des coordonnées des 3 sommets de chaque triangle, puis on assemble les matrices élémentaires pour former les matrices de rigidité et de masse $\mathbb{K}$ et $\mathbb{M}$ (voir codes).
Calculons maintenant le second membre $\vec{L}$. On suppose dans un premier temps que $f = 1$. Comme $1 \in V_h$, on a:
$\forall i \in [\![1, N]\!], (L)_i = \int_\Omega 1 w_i d\Omega = \int_\Omega \sum_{j=1}^N w_i w_j = (\mathbb{M}\vec{1})_i$
Donc $\vec{L} = \mathbb{M}\vec{1}$
Dans le cas général $f \in C^0(\bar{\Omega})$, on approche la donnée f par son interpolation π_hf dans la base $(w_i)_{i \in [1,N]}$, on obtient de même: $\vec{L} = \mathbb{M}\mathbb{F}$ avec $\mathbb{F}_i = f(S_i)$
On admettra pour la suite que quand on remplace f par son interpolée $\pi_h f$ de V_h, l'approximation par des éléments finis P^1 du problème n'est pas altérée.

Enfin, on veut vérifier que le code calcule une solution approchée u_h correcte. Pour cela, on résout le problème (1) avec une solution u connue égale à $u(x, y) = cos(\pi x)cos(2\pi y), pour (x, y) \in \Omega$.
Après calcul de Δu, on injecte le résultat dans la formulation variationnelle, et on obtient $f = (1 + 5\pi^2)u$
En assimilant u à son interpolée $\pi_h u$, on calcule la norme L^2 de l'erreur:
$||u - u_h||^2_{L^2(\Omega)} = ||\pi_h u - u_h||^2_{L^2(\Omega)} = \int_\Omega |\pi_h u - u_h|^2 d\Omega$
$= \int_\Omega |\sum_{i=1}^N (u(S_i) - u_h(S_i))w_i|^2 d\Omega$

$= \int_\Omega \sum_{i=1}^N \sum_{j=1}^N (u(S_i) - u_h(S_i))(u(S_j) - u_h(S_j)) w_i w_j d\Omega$
$= \sum_{i=1}^N \sum_{j=1}^N (u(S_i) - u_h(S_i))(u(S_j) - u_h(S_j))(\mathbb{M})_{ij}$
$= \ ^{\mathsf{T}}(U - U_h)\mathbb{M}(U - U_h)$

On calcule également la semi-norme H^1 de l'erreur:
$|u - u_h|_{H^1(\Omega)}^2 = ||\nabla \pi_h u - \nabla u_h||_{L^2(\Omega)}^2 = \int_\Omega |\nabla \pi_h u - \nabla u_h|^2 d\Omega$
$= \int_\Omega \sum_{i=1}^N \sum_{j=1}^N (u(S_i) - u_h(S_i))(u(S_j) - u_h(S_j)) \nabla w_i \nabla w_j d\Omega$
$= \sum_{i=1}^N \sum_{j=1}^N (u(S_i) - u_h(S_i))(u(S_j) - u_h(S_j))(\mathbb{K})_{ij}$
$= \ ^{\mathsf{T}}(U - U_h)\mathbb{K}(U - U_h)$

On obtient les courbes d'erreur suivantes, la convergence semble être d'ordre 2,2 pour la norme L^2 et d'ordre 1,7 pour la norme H^1.

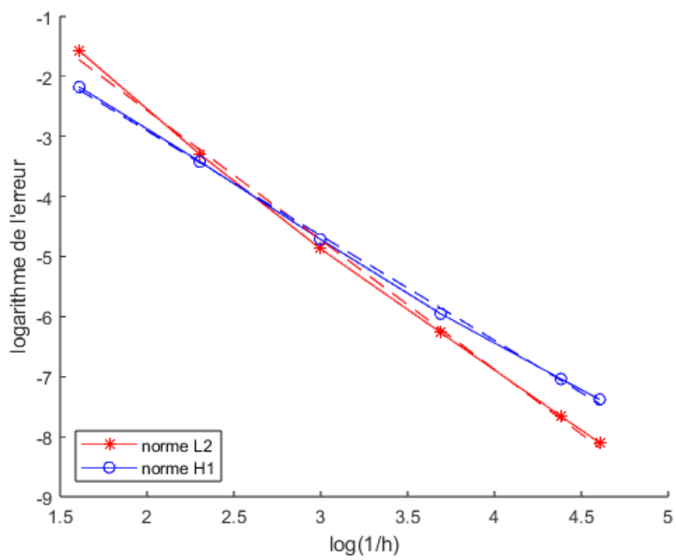

Figure 2: Courbes d'erreur

On tente de résoudre le problème suivant à l'aide de notre code :
Trouver $u \in H^1(\Omega)$ telle que

$$\begin{cases} -\Delta u = f & \text{dans } \Omega \\ \frac{\partial u}{\partial n} = 0 & \text{sur } \partial\Omega \end{cases}$$

On modifie alors notre code Matlab car ici le système linéaire équivalent est $\mathbb{K}\vec{U} = \vec{L}$.
On doit donc inverser $\mathbb{K}$, or $\mathbb{K}$ n'est pas inversible, en effet :
$\mathbb{K}\vec{1} = \left(\sum_{i=1}^{N} \int_\Omega \nabla w_i \cdot \nabla w_j\right)_{1 \leqslant j \leqslant N} = \left(\int_\Omega \nabla \left(\sum_{i=1}^{N} w_i\right) \cdot \nabla w_j\right)_{1 \leqslant j \leqslant N}$
Or $\sum_{i=1}^{N} w_i = 1$ donc $\mathbb{K}\vec{1} = \vec{0}$
Ceci est lié au fait que le système n'admet pas de solution unique, il suffit d'ajouter une constante à une solution pour trouver une nouvelle solution.

Matlab renvoie alors une solution au système matriciel avec des valeurs énormes. Cependant, le vecteur renvoyé est solution du système matriciel et donc la fonction associée est solution du problème approché. Si on veut se rapprocher de la fonction u qui est de moyenne nulle on peut soustraire la moyenne de la solution trouvée pour avoir une solution du problème approché de moyenne nulle. Numériquement il suffit de soustraire au vecteur solution le vecteur ayant pour chaque coordonée la somme des coordonées du vecteur solution, on obtient alors une solution proche de la solution recherchée.