

Résolution numérique d'un problème de Poisson avec des éléments finis P^1

Les sources matlab doivent être récupérées sur le moodle du cours.

En plus du rapport écrit, il faudra envoyer à votre chargé·e pour le 9 octobre 2023 de TD tous les codes et les maillages (dans un dossier compressé), de façon à que les codes puissent être exécutés sans besoin d'intervention et qu'ils renvoient les résultats qui auront été décrits dans le rapport (en particulier, envoyer la routine et tous les fichiers .msh utilisés pour les graphes de convergence). Merci de spécifier si vous avez utilisé Matlab ou Octave.

Soit Ω un ouvert borné à frontière polygonale de \mathbb{R}^2 et $f \in L^2(\Omega)$. On s'intéresse à la résolution numérique du problème de Poisson avec condition aux limites de Neumann :
Trouver $u \in H^1(\Omega)$ telle que

$$(1) \quad \begin{cases} u - \Delta u = f & \text{dans } \Omega \\ \frac{\partial u}{\partial n} = 0 & \text{sur } \partial\Omega \end{cases} .$$

1.1 - Ecrire la formulation variationnelle du problème. Montrer que ce problème est bien posé.

Discrétisation

Soit \mathcal{T}_h une triangulation du domaine Ω , et V_h l'approximation de $H^1(\Omega)$ par des éléments finis P^1 associés à la triangulation \mathcal{T}_h . On note $(T_\ell)_{\ell=1,L}$ les triangles de \mathcal{T}_h , $(S_I)_{I=1,N}$ les sommets des triangles et $(w_I)_{I=1,N}$ la base de V_h définie par $w_I(S_J) = \delta_{IJ}$, $1 \leq I, J \leq N$.

1.2 - Quelle est la formulation variationnelle discrète vérifiée par la solution approchée u_h ?

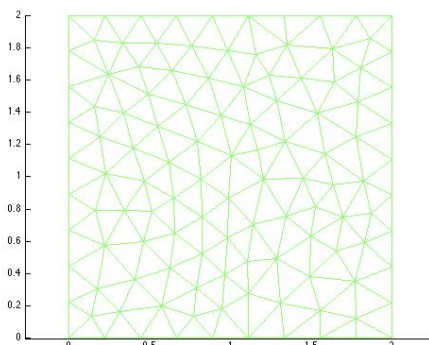
1.3 - Par construction, la solution approchée s'écrit sous la forme

$$u_h(x, y) = \sum_{I=1}^N u_h(S_I) w_I(x, y), \quad \forall (x, y) \in \bar{\Omega}.$$

Exprimer la formulation variationnelle discrète sous la forme d'un système linéaire équivalent :

$$(2) \quad (\mathbb{M} + \mathbb{K}) \vec{U} = \vec{L},$$

de solution le vecteur $\vec{U} \in \mathbb{R}^N$ dont la $I^{\text{ème}}$ composante vaut $u_h(S_I)$. Ci-dessus, \mathbb{M} est la matrice de masse et \mathbb{K} est la matrice de rigidité. Quelles sont les propriétés fondamentales des matrices \mathbb{M} et \mathbb{K} ?

FIGURE 1 – Domaine Ω .

Maillages

On veut résoudre le problème dans un ouvert Ω qui est un carré.

Pour mailler Ω , on choisit le mailleur `Gmsh`. Pour réaliser les calculs, on s'appuie sur `Matlab`. En `Matlab`, la routine `principal_neumann.m` sera le programme principal pour résoudre (2).

1.4 - Pour créer des maillages de l'ouvert carré, voici la procédure à suivre :

- Modifier le fichier `geomCarre.geo` en l'ouvrant avec un éditeur de texte. Par exemple, pour changer le pas du maillage, modifier la valeur de h . Enregistrer les changements.
- Lancer `Gmsh` dans un terminal et ouvrir le fichier `geomCarre.geo`.
- Dans la fenêtre `Gmsh`, menu du haut, choisir “`Mesh`”. Cliquer ensuite sur “`2D`”.
- Pour sauvegarder le maillage ainsi créé, choisir “`Save Mesh`” dans le menu “`File`”.

Un fichier `geomCarre.msh` a été créé.

Attention : le nom du fichier maillage `.msh` est *par défaut* celui du fichier géométrie `.geo`.

1.5 - Afficher le maillage précédemment créé à l'aide de la routine `affichemaillage.m`.

1.6 - Quelle est la signification des structures de données

`Nbpt`, `Nbtri`, `Coorneu`, `Refneu`, `Numtri`, `Reftri`, `Nbaretes`, `Numaretes`, `Refaretes` ?

Les données `Refneu`, `Nbaretes`, `Numaretes`, `Refaretes` ne seront pas utilisées dans ce TP.

Calcul des Matrices élémentaires

Pour calculer les matrices élémentaires sur un triangle, on peut utiliser les formules ci-dessous. Soit T_ℓ un triangle de sommets $S_1(x_1, y_1)$, $S_2(x_2, y_2)$, et $S_3(x_3, y_3)$. Les fonctions

de base locales sont égales aux coordonnées barycentriques λ_1 , λ_2 et λ_3 . En un point (x, y) du triangle, celles-ci sont données par les formules suivantes :

$$\begin{aligned}\lambda_1(x, y) &= \frac{1}{D}(y_{23}(x - x_3) - x_{23}(y - y_3)) \\ \lambda_2(x, y) &= \frac{1}{D}(y_{31}(x - x_1) - x_{31}(y - y_1)) \\ \lambda_3(x, y) &= \frac{1}{D}(y_{12}(x - x_2) - x_{12}(y - y_2))\end{aligned}$$

où $x_{ij} = x_i - x_j$, $y_{ij} = y_i - y_j$ pour i et j différents dans $\{1, 2, 3\}$, et $D = x_{23}y_{31} - x_{31}y_{23}$. Notons que D est égal, au signe près, à deux fois la surface du triangle.

Pour $(k_1, k_2, k_3) \in \mathbb{N}^3$, on a les résultats suivants :

$$\int_{T_\ell} \lambda_1^{k_1} \lambda_2^{k_2} \lambda_3^{k_3} d\Omega = 2 \frac{k_1! k_2! k_3!}{(k_1 + k_2 + k_3 + 2)!} \text{aire}(T_\ell).$$

1.7 - Soit T_ℓ un triangle de sommets $S_1(x_1, y_1)$, $S_2(x_2, y_2)$ et $S_3(x_3, y_3)$. Compléter les routines `matK_elem.m` et `matM_elem.m` qui à partir des coordonnées des 3 sommets d'un triangle donnent respectivement les matrices élémentaires de rigidité et de masse.

Il est conseillé de vérifier le calcul des matrices élémentaires en le comparant à un calcul fait à la main pour le triangle de référence (composé des sommets $(0, 0)$, $(1, 0)$ et $(0, 1)$).

Assemblage des matrices

On rappelle l'algorithme d'assemblage

$\mathbb{M} = 0$, $\mathbb{K} = 0$

Pour $\ell = 1, L$

Détermination des coordonnées des sommets du triangle ℓ

Calcul des contributions élémentaires \mathbb{M}^ℓ , \mathbb{K}^ℓ

Pour $i = 1, 3$

$I = \text{local} \rightarrow \text{global}(\ell, i)$

Pour $j = 1, 3$

$J = \text{local} \rightarrow \text{global}(\ell, j)$

$\mathbb{M}_{IJ} = \mathbb{M}_{IJ} + \mathbb{M}_{ij}^\ell$ et $\mathbb{K}_{IJ} = \mathbb{K}_{IJ} + \mathbb{K}_{ij}^\ell$

Fin pour j

Fin pour i

Fin pour ℓ

1.8 - Compléter la partie assemblage des matrices \mathbb{M} et \mathbb{K} .

Calcul du second membre

1.9 - On suppose dans un premier temps que $f = 1$. En remarquant que $1 \in V_h$, calculer le vecteur second membre \vec{L} à partir de la matrice de masse \mathbb{M} et d'un vecteur calculé dans la routine `f.m`.

1.10 - Rappeler l'expression du second membre \vec{L} pour une donnée générale $f \in C^0(\bar{\Omega})$. En approchant la donnée f par son interpolation $\pi_h f$ dans la base $(w_I)_{I=1,N}$, donner une approximation du second membre \vec{L} faisant intervenir la matrice de masse.

On admettra 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.

Validation

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 $(x, y) \in \bar{\Omega}$.

1.11 - Calculer la donnée f correspondante et modifier la routine `f.m`.

1.12 - En assimilant u à son interpolée $\pi_h u$, donner une estimation de la norme L^2 de l'erreur, $\|u - u_h\|_{L^2(\Omega)}$, faisant intervenir la matrice de masse \mathbb{M} . Comment évolue cette erreur en fonction de h dans le cas de la première géométrie? On pourra tracer $\log(1/h) \mapsto \log(\|u - u_h\|_{L^2(\Omega)}/\|u\|_{L^2(\Omega)})$ pour différentes valeurs de h .

1.13 - En assimilant u à son interpolée $\pi_h u$, donner une estimation de la semi-norme H^1 de l'erreur, $|u - u_h|_1 = \|\nabla u - \nabla u_h\|_{L^2(\Omega)^2}$, faisant intervenir la matrice de rigidité \mathbb{K} . Comment évolue cette erreur en fonction de h dans le cas de la première géométrie? On pourra tracer $\log(1/h) \mapsto \log(|u - u_h|_1/|u|_1)$ pour différentes valeurs de h .

Comment effectuer une étude complète de l'erreur

Tracer un graphe **en échelle log-log** de l'erreur normalisé en norme L^2 , puis en semi-norme H^1 , en fonction du réciproque du pas du maillage h . Plus précisément :

- En abscisse on a $\log(1/h)$ et en ordonnée on a $\log(\|u - u_h\|_{L^2(\Omega)}/\|u\|_{L^2(\Omega)})$, puis $\log(|u - u_h|_{H^1(\Omega)}/|u|_{H^1(\Omega)})$.
- Choisir au moins 3 ou 4 valeurs différentes de h : $h_1 > h_2 > h_3 > h_4$ (souvent on choisit $h_{i+1} = h_i/2$ pour que les points du graphe soient bien espacés).
- Effectuer une régression linéaire (par exemple par la commande `polyfit` en Matlab/Octave) pour calculer les coefficients de la droite de régression.
- Tracer les courbes d'erreur (mettre bien en évidence les 4 points, comme sur la Figure 2), ainsi que les droites de régression (avec un autre style, par exemple en pointillé comme sur la Figure 2).

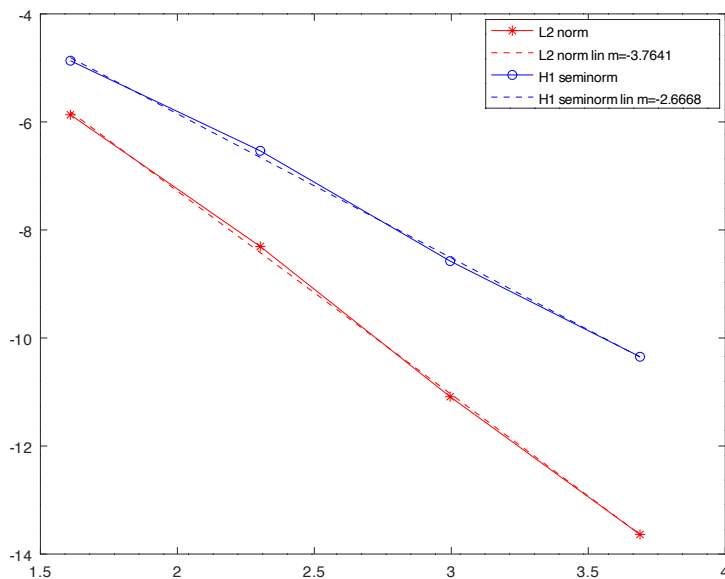


FIGURE 2 – Exemple de graphe de convergence. Attention les ordres de convergence ne sont pas ceux que vous devez obtenir.

Résolution d'un problème similaire

Utiliser votre code pour résoudre le problème suivant

Trouver $u \in H^1(\Omega)$ telle que

$$(3) \quad \begin{cases} -\Delta u = f & \text{dans } \Omega \\ \frac{\partial u}{\partial n} = 0 & \text{sur } \partial\Omega \end{cases} .$$

Que se passe-t-il ? Expliquer.