Projet Approximation des EDP

Feifan PING

April 2025

1 Introduction

Ce projet porte sur la résolution d'une équation aux dérivées partielles elliptique sur un domaine à géométrie complexe à l'aide de la méthode des éléments finis. Le domaine Ω est un carré troué, avec conditions de Dirichlet et Neumann mixtes. Le projet comprend la génération du maillage, la formulation variationnelle du problème, l'assemblage du système linéaire, la résolution numérique, et l'analyse de la convergence.

Il y a trois parties dans ce projet. Dans la première partie, on va donner le domaine et les frontières du problème qu'on considère. Dans la deuxième partie, on va étudier la formation variationnelle du problème EDP et trouver la solution numérique avec la méthode des éléments finis $\mathbb{P}1$ Lagrange. Dans la troisième partie, on va étudier l'erreur entre la solution exacte et la solution numérique par rapport à différente norme, et comparer la vitesse de la convergence.

2 Géométrie du domaine et la frontière

Nous considérons le domaine suivant

$$\Omega := ((0, 2\pi) \times (0, 2\pi)) \backslash ([\pi/2, 3\pi/2] \times [\pi/2, 3\pi/2]),$$

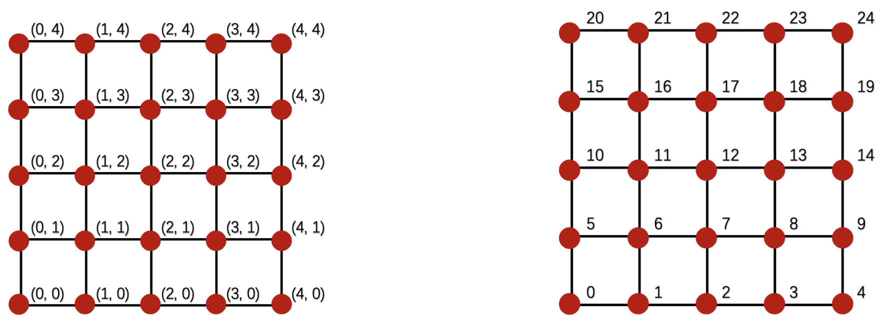
avec la frontière $\Gamma = \Gamma_N \cup \Gamma_D$ désigne la frontière de $(0, 2\pi) \times (0, 2\pi)$ et Γ_D désigne la frontière de $(\pi/2, 3\pi/2) \times (\pi/2, 3\pi/2)$.

Pour la question (a), on va écrire une fonction **GenerateMesh** pour génère un maillage triangulaire structuré uniforme pour Ω:

1.GenerateMesh
· Input: n
· Output: vtx, elt

Ici, n est un paramètre dans $\mathbb{N}$ tel que $N = 4n + 1$ signifie le nombre de points dans chaque direction. On note Ω_D le domaine généré par Γ_D et Ω_N le domaine généré par Γ_N. Cela signifie que $\Omega = \Omega_D \backslash \Omega_N$.

Dans le domaine Ω_N, on peut prendre $(4n+1)^2$ de points qui divisent le domaine en $(4n)^2$ petits carrés. Pour le point (i,j) où $0 \le i,j \le N$, on le donne index $i*N+j$ et le stocke dans une liste qui s'appelle ***points***. Donc on obtient une liste de points avec index de 0 à $(4n+1)^2-1$. Pour chaque petit carré engendré par les quatre points avec coordonnées $(i,j),(i,j+1),(i+1,j),(i+1,j+1), 0 \le i,j \le 4n-1$, on peut le diviser en deux triangles avec coordonnées $[(i,j),(i+1,j),(i+1,j+1)]$ et $[(i,j),(i+1,j+1),(i,j+1)]$. En suite, on stocke touts les triangles dans une liste qui s'appelle ***éléments***. On va expliquer ce processus dans figure 1 et figure 2 avec l'exemple de la condition n=1.



(a) coordonnées

(b) index

Figure 1: Conversion de coordonnées aux index

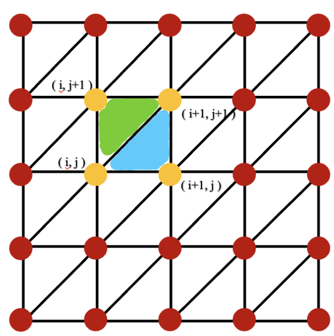
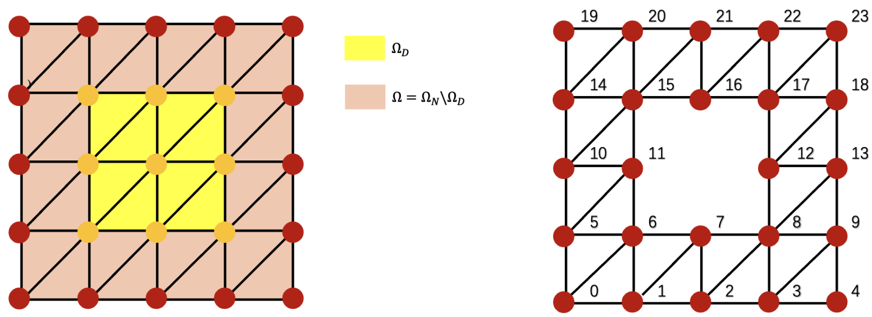


Figure 2: Construction des éléments

Pour construire la zone dont nous avons besoin, nous voulons filtrer les points situés dans Ω_D (Γ_D incluse), ce qu'on va noter ***points_D***. Pour tous les triangles dans la liste ***éléments***, si tous les trois sommets sont dans ***points_D***, nous allons le retirer de la liste ***éléments***. Nous sélectionnons les points dans ***points*** qui sont utilisés pour construire les triangles dans la nouvelle liste ***éléments***. En suite, on va donner les points des nouveaux index. C'est justement la liste des points dans le domaine Ω.



(a) Choix de triangles

(b) Point dans Ω

Figure 3: Façon de construire les éléments et points dans Ω

Pour la question (b) et (c), on va définir deux fonction pour représenter un maillage du domaine Ω et tracer le nouveau maillage et le vecteur unitaire normal sur toute la frontière Γ de Ω.

2.PlotMesh
· Input: vtx, elt
· Output: une image des éléments et points dans le domaine Ω.
3.plot_boundary_normals
· Input: vtx, elt
· Output: une image des vecteurs unitaires normals sur toute la frontière

Il est facile de construire la fonction **PlotMesh** si on réussir de construire la fonction **GenerateMesh**. On peut obtenir le résultat avec n = 1 comme Figure 4:

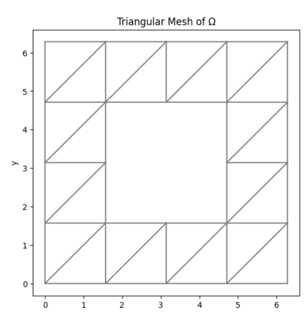
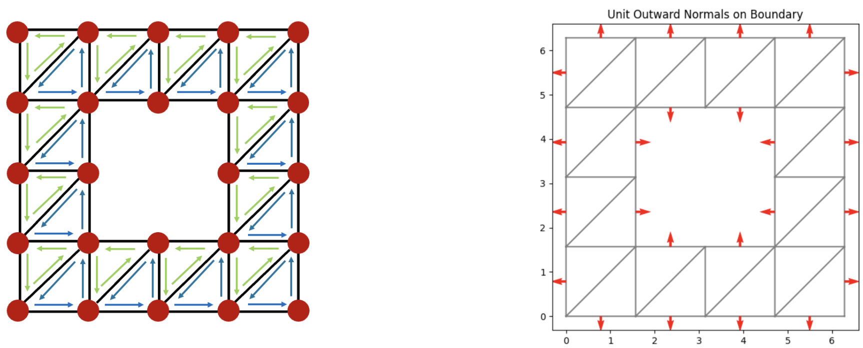


Figure 4: L'image de Ω avec n=1 par python

Puisque tous les sommets des éléments triangulaires sont ordonnés dans le sens anti-horaire, le vecteur normal extérieur de chaque arête pointe vers l'extérieur du triangle. Donc on peut obtenir les images comme Figure 5:



(a) sommets ordonnés

(b) vecteurs normals extérieurs

Figure 5: Construire les vecteurs normals extérieurs

3 Problème d'EDP

Dans cette partie, nous voulons considère un problème d'EDP et trouver la solution numérique.
Nous considérons le modèle d'EDP du second ordre suivant:

$$\begin{cases} Trouver\ u \in H^1(\Omega)\ tel\ que : & \\ -\Delta u + \mu u = f, & dans\ \Omega \\ u = 0, & sur\ \Gamma_D \\ \partial_n u = \partial_n u_{ex}, & sur\ \Gamma_N. \end{cases}$$

Où les fonctions utilisées sont définies comme suivant: Soit

$$\mu(x,y) := \begin{cases} 1, & si\ y < \pi, \\ 2, & si\ y > \pi. \end{cases}$$

Soit $p,\ q \in \mathbb{N}$,

$$f(x,y) := (4(p^2+q^2)+\mu) sin(2px) sin(2qy), \qquad \forall (x,y) \in \Omega,$$

$$u_{ex}(x,y) := sin(2px) sin(2qy), \qquad \forall (x,y) \in \Omega.$$

I. Formulation variationnelle

On se donne un espace $V = \{v \in H^1(\Omega) \mid v = 0 \ sur \ \Gamma_D\}$. Pour tout $v \in V$, on a que:

$$\int_\Omega (-\Delta u\overline{v} + \mu u\overline{v})dx = \int_\Omega f\overline{v} \ dx$$

$$\int_\Omega -div(\nabla u)\overline{v} \ dx \ + \ \int_\Omega \mu u\overline{v} \ dx \ = \ \int_\Omega f\overline{v} \ dx$$

Puisque $div(\nabla u \cdot \overline{v}) = div(\nabla u)\overline{v} \ + \ \nabla u \cdot \nabla v$, on obtient que

$$\int_\Omega \nabla u\nabla v \ dx \ - \ \int_\Omega div(\nabla u \cdot \overline{v}) \ dx \ + \ \int_\Omega \mu u\overline{v} \ dx \ = \ \int_\Omega f\overline{v} \ dx$$

Par formule de stokes, on a que

$$\int_\Omega \nabla u\nabla v \ dx \ - \ \int_{\partial\Omega} \nabla u \cdot \overline{v} \cdot \mathbf{n} \ d\sigma \ + \ \int_\Omega \mu u\overline{v} \ dx \ = \ \int_\Omega f\overline{v} \ dx$$

$$\int_\Omega \nabla u\nabla v \ dx \ - \ \int_{\Gamma_N} \partial_\mathbf{n} u \cdot \overline{v} \ d\sigma \ - \ \int_{\Gamma_D} \nabla u \cdot \overline{v} \cdot \mathbf{n} \ d\sigma \ + \ \int_\Omega \mu u\overline{v} \ dx \ = \ \int_\Omega f\overline{v} \ dx$$

Par condition au bord, on a que

$$\partial_\mathbf{n} u = \partial_\mathbf{n} u_{ex} \qquad\qquad sur \ \Gamma_N$$

$$v = 0 \ \ sur \ \Gamma_D \quad pour \ tout \ v \in V$$

Donc on peut obtenir que

$$\int_\Omega \nabla u\nabla v \ dx \ - \ \int_{\Gamma_N} \partial_\mathbf{n} u_{ex}\overline{v} \ d\sigma \ + \ \int_\Omega \mu u\overline{v} \ dx \ = \ \int_\Omega f\overline{v} \ dx$$

Finalement, on obtient la formulation variationnelle $a(u, v) = l(v)$ où

$$a(u, v) = \int_\Omega \nabla u\nabla v \ dx \ + \ \int_\Omega \mu u\overline{v} \ dx$$

$$l(v) = \int_\Omega f\overline{v} \ dx \ + \ \int_{\Gamma_N} \partial_\mathbf{n} u_{ex}\overline{v} \ d\sigma$$

II. Matrice locale et globale

Notre objet suivant est de construire le système linéaire de ce problème. Par la formulation variationnelle, on peut obtenir la formule matricielle:

$$(K + M)u = b_f + b_N$$

où K est la matrice de rigidité associée à $\int_\Omega \nabla u \cdot \nabla v \ dx$ et M est la matrice de masse associée à $\int_\Omega \mu u\overline{v} \ dx$. Pour construire la matrice globale, on va tout d'abord considérer la matrice de rigidité et la matrice de masse locale pour un élément triangulaire. C'est justement ce qu'on va faire dans question (b).

(1) Étape 1: Calculer la matrice de rigidité

On va définir une fonction **Kloc** pour construire la matrice de rigidité:

4.Kloc
· Input: vtx, e(les index des sommets de l'élément triangulaire)
· Output: K(la matrice de rigidité locale)

Soit $p_0,\ p_1,\ p_2$ sont les trois sommets d'un élément triangulaire qui sont donnés par:

$$p_0 := (x_0,\ y_0), \qquad p_1 := (x_1,\ y_1), \qquad p_2 := (x_2,\ y_2)$$

Donc l'aire de ce domaine $|s|$ est calculé par

$$|s| = \frac{1}{2}|\mathbf{p_0p_1} \times \mathbf{p_0p_2}| = \frac{1}{2}|(x_1 - x_0)(y_2 - y_0) - (x_2 - x_0)(y_1 - y_0)|$$

En suite, on va considère les fonctions de base sur ces trois points. Pour la méthode des éléments finis $\mathbb{P}_1$ Lagrange, les fonctions de bases $\phi_0,\ \phi_1,\ \phi_2$ sont des fonctions linéaires définies par:

$$\phi_i(p_j) = \begin{cases} 1, & si \ \ i = j \\ 0. & si \ \ i \neq j \end{cases}$$

Où $i,\ j = 0,\ 1,\ 2$. On suppose que $\phi_i(x,\ y)$ a la forme

$$\phi_i(x,\ y) := a_i x + b_i y + c_i$$

On se donne l'exemple de condition $i = 0$.Par la définition de la fonction de base, on peut obtenir un système d'équations linéaires:

$$\begin{cases} a_0x_0 + b_0y_0 + c_0 = 1 \\ a_0x_1 + b_0y_1 + c_0 = 0 \\ a_0x_2 + b_0y_2 + c_0 = 0 \end{cases}$$

Où $(x_0,\ y_0),\ (x_1,\ y_1),\ (x_2,\ y_2)$ sont des coordonnées connues. On peut résoudre ce système d'équations linéaires et obtenir le résultat:

$$\begin{cases} a_0 = \frac{y_2 - y_1}{(x_0 - x_1)(y_2 - y_1) - (x_2 - x_1)(y_0 - y_1)} = \frac{1}{2|s|}(y_2 - y_1) \\ b_0 = \frac{x_1 - x_2}{(x_0 - x_1)(y_2 - y_1) - (x_2 - x_1)(y_0 - y_1)} = \frac{1}{2|s|}(x_1 - x_2) \\ c_0 = \frac{x_2y_1 - x_1y_2}{(x_0 - x_1)(y_2 - y_1) - (x_2 - x_1)(y_0 - y_1)} = \frac{1}{2|s|}(x_2y_1 - x_1y_2) \end{cases}$$

Pour le point p_i, le gradient de la fonction de base ϕ_i est donné par

$$\nabla\phi_i(x,\ y) = \begin{pmatrix} a_i \\ b_i \end{pmatrix} = \frac{1}{2|s|}\begin{pmatrix} y_j - y_k \\ x_k - x_j \end{pmatrix}$$

Où $j,\ k = 0,\ 1,\ 2$ et $j,\ k \neq i$. On peut trouver que sur l'élément triangulaire, la fonction de base ϕ_i pour la méthode des éléments finis $\mathbb{P}_1$ est justement un surface passant les points $(p_i,\ 1)$, $(p_j,\ 0)$, $(p_k,\ 0)$ et le gradient associé est justement un vecteur constant en surface x-y pointant dans la direction de la montée la plus rapide de la fonction.

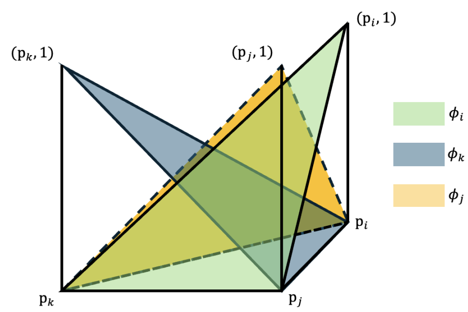


Figure 6: Les fonctions de base sur un élément triangulaire

Donc pour la matrice de rigidité locale, on a que:

$$K_{i,j}^{(s)} = \int_s \nabla\phi_i \cdot \nabla\phi_j\ dx = (\nabla\phi_i)^T \nabla\phi_j \cdot |s|$$

(2) Étape 2: Calculer la matrice de masse

En suite, on va calculer la matrice de masse associée à $\int_\Omega \mu u\overline{v}\ dx$. On définit une fonction **Mloc_mu** pour le construire:

5.Mloc_mu
· Input: vtx, e(les index des sommets de l'élément triangulaire)
· Output: M(la matrice de masse locale)

Soient $\phi_0,\ \phi_1,\ \phi_2$ les fonctions de base pour le domaine triangulaire, $M_{i,j}$ est calculée par

$$M_{i,j} = \int_s \mu\phi_i\phi_j dxdy$$

Pour calculer cet intégrale, on va tout d'abord considère le triangle spécial $\hat{s}$:

$$\hat{s} := \{(\xi,\ \eta) \in \mathbb{R}^2 | \xi \geq 0,\ \eta \geq 0,\ \xi + \eta \leq 1\}$$

Les trois sommets de $\hat{s}$ sont $\hat{p_0} = (0,\ 0)$, $\hat{p_1} = (1,\ 0)$ et $\hat{p_2} = (0,\ 1)$. Sur l'élément triangle $\hat{s}$, on peut définir les fonctions de base spéciales:

$$\begin{cases} \hat{\phi}_0(\xi,\ \eta) = 1 - \xi - \eta \\ \hat{\phi}_1(\xi,\ \eta) = \xi \\ \hat{\phi}_2(\xi,\ \eta) = \eta \end{cases}$$

On peut voir que ces trois fonctions vérifient les conditions de fonctions de base pour la méthode des éléments finis $\mathbb{P}_1$ Lagrange (linéaire, $\hat{\phi}_i(\hat{p_i}) = 1$ *et* $\hat{\phi}_i(\hat{p_j}) = 0$ si $i \neq j$). Notre idée est de définir une fonction affine $F : \hat{s} \to s$ telle que:

$$F(0,0) = p_0, \quad F(1,0) = p_1, \quad F(0,1) = p_2$$

Pour vérifier cette condition, on peut donc définir la forme de F comme suivant:

$$F(\xi,\ \eta) := p_0 + \xi(p_1 - p_0) + \eta(p_2 - p_0)$$

Soit $p = (x, y) \in s$, on peut l'écrire sous la forme

$$p = \begin{pmatrix} x(\xi,\ \eta) \\ y(\xi,\ \eta) \end{pmatrix} = F(\xi,\ \eta) = \begin{pmatrix} x_0 \\ y_0 \end{pmatrix} + \begin{pmatrix} x_1 - x_0 \\ y_1 - y_0 \end{pmatrix} \xi + \begin{pmatrix} x_2 - x_0 \\ y_2 - y_0 \end{pmatrix} \eta$$

Et on peut obtenir deux fonctions:

$$x(\xi,\ \eta) = x_0 + (x_1 - x_0)\xi + (x_2 - x_0)\eta$$

$$y(\xi,\ \eta) = y_0 + (y_1 - y_0)\xi + (y_2 - y_0)\eta$$

Donc on peut calculer la matrice jacobienne de F, en notant J:

$$J = \begin{pmatrix} \frac{\partial x}{\partial \xi} & \frac{\partial x}{\partial \eta} \\ \frac{\partial y}{\partial \xi} & \frac{\partial y}{\partial \eta} \end{pmatrix} = \begin{pmatrix} x_1 - x_0 & x_2 - x_0 \\ y_1 - y_0 & y_2 - y_0 \end{pmatrix}$$

$$|det(J)| = |(x_1 - x_0)(y_2 - y_0) - (x_2 - x_0)(y_1 - y_0)| = 2|s|$$

On peut calculer M_{ij} par:

$$\int_s \mu\phi_i(x,y)\phi_j(x,y)\ dxdy = |det(J)| \int_{\hat{s}} \mu\hat{\phi_i}(\xi,\eta)\hat{\phi_j}(\xi,\eta)\ d\xi d\eta$$

Nous rappelons que μ est une fonction constante par morceaux:

$$\mu(x,y) := \begin{cases} 1, & si\ y < \pi, \\ 2, & si\ y > \pi. \end{cases}$$

Pour un certain élément triangulaire dans Ω, nous observons qu'il est contenu dans la région $y < \pi$ ou la région $y > \pi$ complètement. Il n'y a pas d'élément triangulaire qui est passé par la ligne $y = \pi$. Donc pour le calcul de matrice de masse locale, μ est un constant en fait, ce qu'on note μ_s:

$$\mu_s = \begin{cases} 1 & si\ \ \frac{1}{3}(y_0 + y_1 + y_2) < \pi \\ 2 & si\ \ \frac{1}{3}(y_0 + y_1 + y_2) > \pi \end{cases}$$

On va calculer les intégrales dans $\hat{s}$:

$$\int_{\hat{s}} \hat{\phi}_i^2(\xi,\ \eta)\ d\xi d\eta = \begin{cases} \int_0^1 \int_0^{1-\xi} (1 - \xi - \eta)^2\ d\eta d\xi = \frac{1}{12} & si\ i = 0 \\ \int_0^1 \int_0^{1-\xi} \xi^2\ d\eta d\xi = \frac{1}{12} & si\ i = 1 \\ \int_0^1 \int_0^{1-\eta} \eta^2\ d\xi d\eta = \frac{1}{12} & si\ i = 2 \end{cases}$$

$$\int_{\hat{s}} \hat{\phi}_i(\xi,\ \eta)\hat{\phi}_j(\xi,\ \eta)\ d\xi d\eta = \frac{1}{24} \quad si\ \ i \neq j$$

Finalement, on obtient la matrice de masse locale dans l'élément triangulaire s:

$$M^{(s)} = 2\mu_s|s| \cdot \begin{pmatrix} \frac{1}{12} & \frac{1}{24} & \frac{1}{24} \\ \frac{1}{24} & \frac{1}{12} & \frac{1}{24} \\ \frac{1}{24} & \frac{1}{24} & \frac{1}{12} \end{pmatrix} = \frac{\mu_s|s|}{12} \cdot \begin{pmatrix} 2 & 1 & 1 \\ 1 & 2 & 1 \\ 1 & 1 & 2 \end{pmatrix}$$

(3) Étape 3: Calculer la matrice globale

Maintenant, on a déjà la matrice de rigidité $K^{(s)}$ et la matrice de masse $M^{(s)}$ pour chaque élément triangulaire. On va écrire la fonction **Matrice_globale** pour calculer la matrice complète du système linéaire associé.

6.Matrice_globale
· Input: vtx, elt
· Output: A(la matrice complète)

Tout d'abord, puisque la dimension de la matrice globale A est associée au nombre de points dans le domaine Ω, donc on pose **m** qui signifie la longueur de la liste **vtx**. Nous parcourons chaque élément triangulaire et calcule les matrices de rigidité et masse associées.

Ensuite, nous allons assembler la matrice globale à l'aide de la méthode scipy.sparse.coo_matrix. Pour utiliser cette fonction, nous avons besoins les paramètres suivants:

· I: Indice de ligne des éléments dans la matrice creuse
· J: Indice de colonne des éléments dans la matrice creuse
· K_s: Les valeurs de la matrice de rigidité
· M_s: Les valeurs de la matrice de masse

Soit un élément **e** de la liste **elt**, il est un tableau à trois éléments qui signifie les index globals dans la liste **vtx**. Donc le résultat de la fonction ***enumerate(e)*** retourne deux tableaux: le premier signifie les index des sommets de l'élément triangulaire locale et le deuxième signfie les index globals dans la liste **vtx**.

On suppose les index des trois sommets d'un élément triangulaire s sont donnés par:
p_0: local 0 global i, p_1: local 1 global j, p_2: local 2 global k
On ajoute les index globals dans la liste I et J. Alors les matrices de rigidité et masse locales s'écrivent comme suivant:

$$K^s = \begin{pmatrix} K^s_{00} & K^s_{01} & K^s_{02} \\ K^s_{10} & K^s_{11} & K^s_{12} \\ K^s_{20} & K^s_{21} & K^s_{22} \end{pmatrix} \quad et \quad M^s = \begin{pmatrix} M^s_{00} & M^s_{01} & M^s_{02} \\ M^s_{10} & M^s_{11} & M^s_{12} \\ M^s_{20} & M^s_{21} & M^s_{22} \end{pmatrix}$$

Chaque élément dans K^s et M^s contribue sa propre force au point correspondant, cela signifie que

$$\begin{aligned}
&K[i,\ i]+=K^s_{00}\\
&K[i,\ j]+=K^s_{01}\\
&.........\\
&K[k,\ k]+=K^s_{22}
\end{aligned}$$

au total 9 fois. Même calcul pour la matrice de masse globale. Pour le construire par python, on ajoute les valeurs locales dans la liste K_s et M_s. À l'aide de coo_matrix on peut obtenir les matrices de rigidité globale K et masse globale M. Finalement, on obtenir la matrice complète A de dimension $\mathbf{m} \times \mathbf{m}$ par:

$$A = K + M$$

III. Approximation de $l(v)$

Nous rappelons la formation de $l(v)$:

$$l(v) = \int_{\Gamma_N} \partial_{\mathbf{n}} u_{ex} \overline{v}\ d\sigma + \int_{\Omega} f \overline{v}\ dx$$

On observe qu'il existe deux parties dans la formulation de $l(v)$, la première associée à la condition au bord et la deuxième associée à la fonction f.

(1) Étape 1: Terme de condition frontière de Neumann

On observe que la frontière Γ_N a quatre côtés, donc on va faire le calcul d'intégral à quatre côtés. Nous devons d'abord faire quelques préparatifs:
Premièrement, calculer ∇u_{ex}.

Puisque on a que

$$u_{ex}(x,\ y) = sin(2px) \cdot sin(2qy)$$

Donc on peut obtenir que

$$\nabla u_{ex}(x,\ y) = \begin{pmatrix} \frac{\partial u_{ex}}{\partial x} \\ \frac{\partial u_{ex}}{\partial y} \end{pmatrix} = \begin{pmatrix} 2p\ cos(2px) \cdot sin(2qy) \\ 2q\ sin(2px) \cdot cos(2qy) \end{pmatrix}$$

Et on a la relation

$$\partial_n u_{ex} = \nabla u_{ex} \cdot \mathbf{n}$$

Deuxièmement, calculer le vecteur normal unitaire extérieur n.

Pour la frontière Γ_N, il y a quatre sommets qu'on note $P_1 = (0,\ 0),\ P_2 = (2\pi,\ 0),\ P_3 = (2\pi, 2\pi),\ P_4 = (0,\ 2\pi)$. Les quatre côtés peuvent être représentés sous la forme suivante:

$$\begin{gathered} L_1 = P_2 - P_1 = (2\pi,\ 0) \\ L_2 = P_3 - P_2 = (0,\ 2\pi) \\ L_3 = P_4 - P_3 = (-2\pi,\ 0) \\ L_4 = P_1 - P_4 = (0,\ -2\pi) \end{gathered}$$

Les vecteurs normals dans le sens antihoraire sont donnés par

$$\begin{gathered} \hat{\mathbf{n}}_1 = (0,\ -2\pi) \\ \hat{\mathbf{n}}_2 = (2\pi,\ 0) \\ \hat{\mathbf{n}}_3 = (0,\ 2\pi) \\ \hat{\mathbf{n}}_4 = (-2\pi,\ 0) \end{gathered}$$

Ensuite, nous devons normaliser les vecteurs:

$$\mathbf{n} = \frac{\hat{\mathbf{n}}}{||\hat{\mathbf{n}}||}$$

Les résultats sont données par $\mathbf{n}_1 = (0,\ -1),\ \mathbf{n}_2 = (1,\ 0),\ \mathbf{n}_3 = (0,\ 1),\ \mathbf{n}_4 = (-1,\ 0)$. Et on peut calculer $\partial_n u_{ex}$ sur les quatre arêtes de Γ_N.

$$\partial_n u_{ex} = \begin{cases} -2q \cdot sin(2px), & sur\ L_1\ car\ y = 0 \\ 2p \cdot sin(2qy), & sur\ L_2\ car\ x = 2\pi \\ 2q \cdot sin(2px), & sur\ L_3\ car\ y = 2\pi \\ -2p \cdot sin(2qy) & sur\ L_4\ car\ x = 0 \end{cases}$$

Troisièmement, calculer l'intégral.

On rappelle la formulation du système linéaire:

$$A \cdot U = b = b_f + b_N$$

Où A est une matrice de dimension $m \times m$ et b un vecteur de dimension $m \times 1$. (m est la longueur de la liste **vtx** qui signifie le nombre des points dans Ω). Pour chaque composante de b_N, on a la formulation:

$$b_i^s = \int_{\Gamma_N} \partial_n u_{ex} \cdot \phi_i \ d\sigma$$

Où ϕ_i est la fonction de base sur le point P_i. Grâce à la condition au bord, si $P_i \notin \Gamma_N$, on a que $b_i^s = 0$. Donc il suffit de ne considérer que les fonctions de base associées aux points situés sur le bord.

Pour une arête du bord, les sommets d'élément triangulaire la divisent en $4n$ parties et la longueur de chaque petit segment est égale à $\frac{\pi}{n}$. D'abord on considère une petit segment AB. On va faire l'intégral pour les deux fonctions de base:

$$\begin{gathered} A : \int_{AB} \partial_n u_{ex}(x, y) \cdot \phi_A(x, y)\ d\sigma \\ B : \int_{AB} \partial_n u_{ex}(x, y) \cdot \phi_B(x, y)\ d\sigma \end{gathered}$$

Pour un point donné, le résultat de l'intégrale $b_i^s = \int_{\Gamma_N} \partial_n u_{ex} \cdot \phi_i \, d\sigma$ ne dépend que des intégrales sur les segments dont ce point est une extrémité. On peut diviser les points situés sur le bord en deux catégories : les sommets du carré et les autres points situés uniquement sur un seul côté.

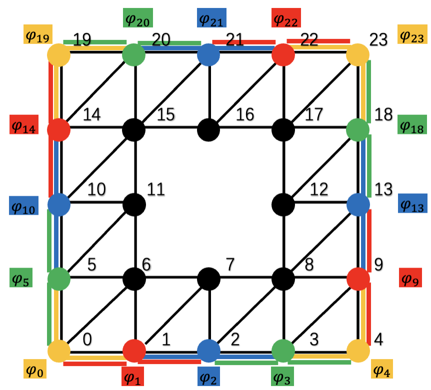


Figure 7: fonctions de base au bord avec l'exemple n=1

On va définir une fonction pour assembler b_N:

7. assemble_b_N
· Input: n, g_N(la fonction pour calculer $\partial_n u_e x$ sur le point (x,y) au bord), p, q
· Output: b_N

(2) Étape 2: Approximation numérique du terme associé à f

Maintenant, on va calculer l'approximation numérique de l'intégrale associée à la fonction f: $\int_\Omega f(x,y)\phi_i(x,y)\,dxdy$. Dans le système linéaire, b_f est un vecteur de dimension m.

Tout d'abord, on va considérer ce problème en chaque élément triangulaire. Soient A, B, C les sommets d'élément triangulaire s, et l'aire du triangle est égale à $|s|$. Soit G le centre de gravité du triangle. Puisque ϕ_A, ϕ_B, ϕ_C sont linéaires telles que $\phi_i(P_j) = \delta_{i,j}$, donc on peut obtenir que

$$b_{i,s}^f = \int_s f\phi_i \, dx \approx f(G) \cdot \phi_i(G) \cdot |s|$$

Grâce à la linéarité de la fonction de base ϕ, $\phi(G) = \frac{1}{3}$ et on a

$$b_{i,s}^f = \int_s f\phi_i \, dx \approx \frac{1}{3} \cdot f(G) \cdot |s|$$

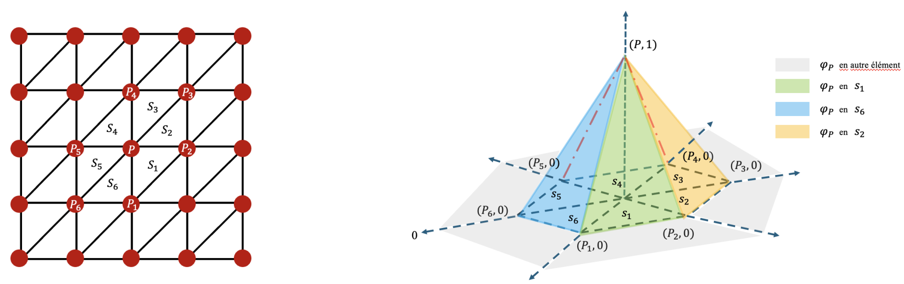


Figure 8: fonction de base sur un point: ϕ_P

Nous appelons cela la contribution d'élément s à l'intégrale globale. Pour calculer $b_i^f = \int_\Omega f \cdot \phi_i \, dx$, on devons additionner les contributions provenant de chaque triangle contenant ϕ_i. On va aussi définir une solution pour assembler u_f:

8. assemble_b_f
· Input: n, f, p, q
· Output: b_f

4 Résolution

Dans cette partie, notre objectif est d'étudier l'erreur entre la solution numérique et la solution exacte, et d'analyser la convergence de la méthode des éléments finis.

I. Montrer que u_{ex} est la solution exacte du problème.
Puisque on a:

$$u_{ex}(x,\ y) = sin(2px) \cdot sin(2qy)$$

$$\Delta u_{ex} = \frac{\partial^2 u_{ex}}{\partial x^2} + \frac{\partial^2 u_{ex}}{\partial y^2}$$

Où

$$\frac{\partial u_{ex}}{\partial x} = 2p \cdot cos(2px)sin(2qy) \ \rightarrow \ \frac{\partial^2 u_{ex}}{\partial x^2} = -4p^2 \cdot sin(2px)sin(2qy)$$
$$\frac{\partial u_{ex}}{\partial_y} = 2q \cdot sin(2px)cos(2qy) \ \rightarrow \ \frac{\partial^2 u_{ex}}{\partial y^2} = -4q^2 \cdot sin(2px)sin(2qy)$$
$$\Delta u_{ex} = -4(p^2 + q^2) \cdot sin(2px)sin(2qy)$$

Nous remplaçons les résultats dans l'équation aux dérivées partielles:

$$-\Delta u_{ex} + \mu u_{ex} = 4(p^2+q^2) \cdot sin(2px)sin(2qy) + \mu(x,\ y)sin(2px)sin(2qy) = f(x, y)$$

Donc u_{ex} est justement la solution exacte du problème.

II. Donner la solution numérique du problème.

Pour $\forall v_h \in V_h$, elle se décompose sur la base $(\phi_i)_{0\leq i\leq m-1}$ (m est la longueur de la liste vtx). On nôte $u_h = \Sigma_{i=0}^{m-1}\lambda_i\phi_i$. Calculer u_h est équivalente à calculer ses composantes λ_i. On notera $U_h = (\lambda_0,\ \lambda_1, ..., \lambda_{m-1})$. Par théorème de Lax-Milgram, U_h est la unique solution du système linéaire $AU_h = b_N + b_f$. Donc on peut résoudre ce système linéaire avec les résultats précédents à l'aide de np.linalg.solve.

On définit une fonction pour donner l'image de la solution numérique:

9. PlotApproximation
· Input: vtx, elt, uh, title
· Output: une image qui précise la solution numérique sur Ω.

On peut obtenir le résultat suivant:

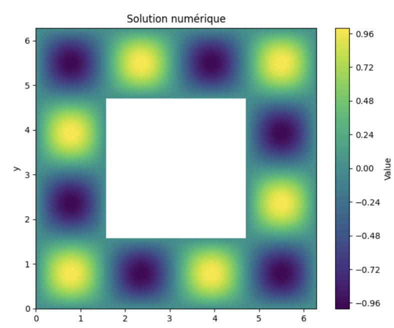


Figure 9: Solution numérique sur Ω

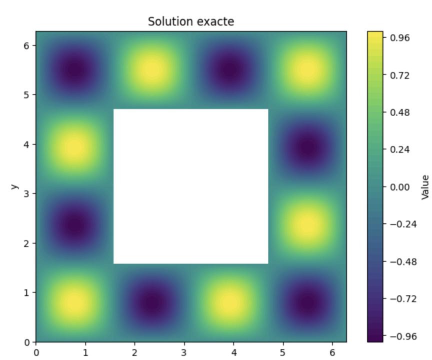


Figure 10: Solution numérique sur Ω

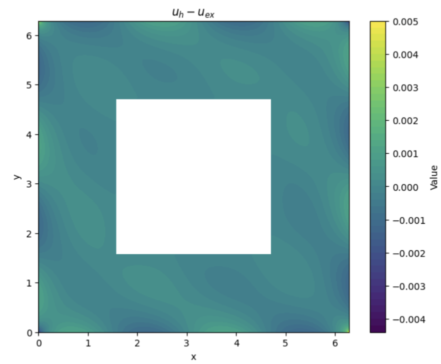


Figure 11: Erreur entre Solution numérique et exacte sur Ω

En suite, on rappelle la formulation d'interpolation de $u_h \in V_h$ locale dans un élément triangulaire s_i:

$$\Pi_i^h u = u_{iA}\phi_A + u_{iB}\phi_B + u_{iC}\phi_C$$

Où $A(x_0, y_0),\ B(x_1, y_1),\ C(x_2, y_2)$ sont les trois sommets de l'élément triangulaire. Comme nous avons calculé avant, les bases locals s'expriment:

$$\begin{aligned}\phi_A(x,y) &= a_0 x + b_0 y + c_0\\ \phi_B(x,y) &= a_1 x + b_1 y + c_1\\ \phi_C(x,y) &= a_2 x + b_2 y + c_2\end{aligned}$$

Où les coefficients sont données par:

$$\begin{aligned}a_0 &= \frac{y_2 - y_1}{2|s|}, \quad b_0 = \frac{x_1 - x_2}{2|s|}, \quad c_0 = \frac{x_2 y_1 - x_1 y_2}{2|s|}\\ a_1 &= \frac{y_0 - y_2}{2|s|}, \quad b_1 = \frac{x_2 - x_0}{2|s|}, \quad c_1 = \frac{x_0 y_2 - x_2 y_0}{2|s|}\\ a_2 &= \frac{y_1 - y_0}{2|s|}, \quad b_2 = \frac{x_0 - x_1}{2|s|}, \quad c_2 = \frac{x_1 y_0 - x_0 y_1}{2|s|}\end{aligned}$$

On peut aussi calculer les gradient des bases:

$$\begin{aligned}\nabla\phi_A(x,y) &= (a_0,\ b_0)^T\\ \nabla\phi_B(x,y) &= (a_1,\ b_1)^T\\ \nabla\phi_C(x,y) &= (a_2,\ b_2)^T\end{aligned}$$

Donc on peut obtenir que

$$\nabla(\Pi_i^h u) = u_{iA}\nabla\phi_A + u_{iB}\nabla\phi_B + u_{iC}\nabla\phi_C$$

Dans ce problème, on pose $h = \frac{\sqrt{2}\pi}{2n}$ la longueur de l'hypoténuse du triangle. Donc l'aire de l'élément triangulaire est égale:

$$|s| = \frac{h^2}{4}$$

Et l'erreur globale par rapport à la norme L^2 est donnée par

$$||u - \Pi_h u||^2_{L^2(\Omega)} = \Sigma_{i=0}^{m-1} ||u - \Pi_h u||^2_{L^2(s_i)}$$

Il est convergent d'ordre 2.

$$||u - \Pi_h u||^2_{H^1(\Omega)} = \Sigma_{i=0}^{m-1} ||u - \Pi_h u||^2_{H^1(s_i)}$$

Il est convergent d'ordre 1.
Par python, nous pouvons obtenir les images suivantes:

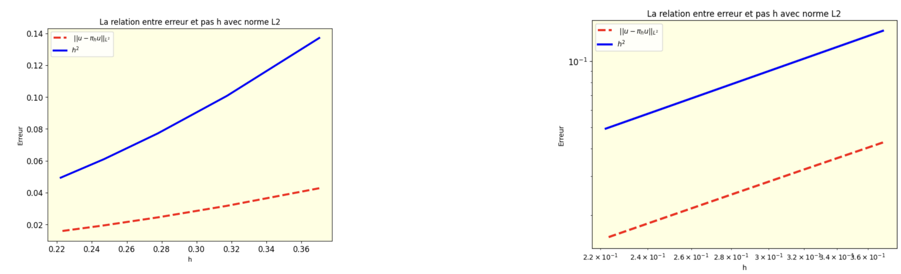


Figure 12: l'erreur pour la norme L2

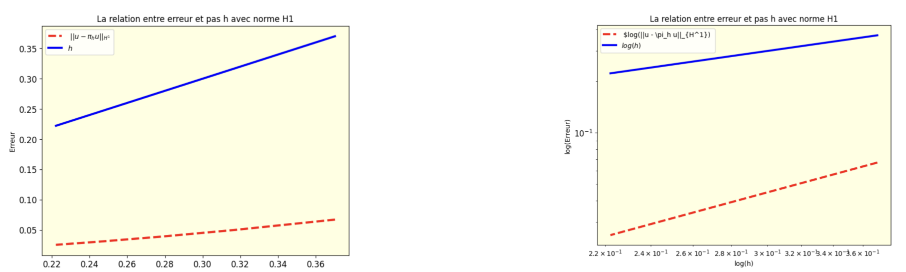


Figure 13: l'erreur pour la norme H1