

Rapport TP1 ANN201

Résolution numérique d'un problème de poisson avec des éléments finis

Tom CUEL

October 20, 2024



Contents

1	Formulation du variationnelle	3
2	Discrétisation	4
3	Maillages	6
4	Matrice élémentaires	6
5	Assemblage des matrices élémentaires	9
6	Calcul du second membre	10
6.1	Calcul théorique	10
6.2	Vérification avec $f = 1$	10
7	Validation	11
8	Résolution d'un problème similaire	14

1 Formulation du variationnelle

Soit Ω un ouvert borné à frontière polygonale ("régulière") 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 :

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

Pour trouver la formulation variationnelle, on multiplie par une fonction test $v \in H^1(\Omega)$ et on intègre sur Ω :

$$\int_{\Omega} uv \, d\Omega - \int_{\Omega} \Delta u v \, d\Omega = \int_{\Omega} f v \, d\Omega.$$

Cependant, $u \in H^1(\Omega)$ et $\Delta u = u + f \in L^2(\Omega)$, donc $u \in H^1(\Omega, \Delta)$, on peut donc supposer $u \in H^2(\Omega)$ et utiliser la formule de Green :

$$\forall v \in H^1(\Omega), \int_{\Omega} (uv + \nabla u \cdot \nabla v) \, d\Omega + \int_{\partial\Omega} (\nabla u \cdot \vec{n} v) \, d\Gamma = \int_{\Omega} f v \, d\Omega \quad .$$

Or, on a $\nabla u \cdot \vec{n} = 0$ sur $\partial\Omega$, donc :

$$\boxed{\text{Trouver } u \in H^1(\Omega) \text{ telle que } \forall v \in H^1(\Omega), \int_{\Omega} (uv + \nabla u \cdot \nabla v) \, d\Omega = \int_{\Omega} f v \, d\Omega} \quad (2)$$

Pour montrer que le problème est bien posé, on va utiliser le théorème de Lax-Milgram. Vérifions ses hypothèses :

Premièrement, on a $H^1(\Omega)$ est un espace de Hilbert.

On définit :

$$a : \begin{array}{ccc} H^1(\Omega) \times H^1(\Omega) & \longrightarrow & \mathbb{R} \\ (u, v) & \longmapsto & \int_{\Omega} (uv + \nabla u \cdot \nabla v) \, d\Omega \end{array}$$

a est clairement bilinéaire et continue (*preuve ci dessous par Cauchy-Schwarz*).

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

De plus, a est coercive :

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

On définit la forme linéaire :

$$l : \begin{array}{ccc} H^1(\Omega) & \longrightarrow & \mathbb{R} \\ v & \longmapsto & \int_{\Omega} f v \, d\Omega \end{array}$$

l est clairement linéaire et continue (*preuve ci dessous par Cauchy-Schwarz*).

$$\begin{aligned} |l(v)| &\leq \|f\|_{L^2(\Omega)} \cdot \|v\|_{L^2(\Omega)} \\ &\leq \|f\|_{L^2(\Omega)} \cdot \|v\|_{H^1(\Omega)}. \end{aligned}$$

On a donc bien vérifié les hypothèses du théorème de Lax-Milgram, le problème est donc bien posé.

2 Discrétisation

Soit T_h une triangulation du domaine Ω et V_h l'approximation de $H^1(\Omega)$ par éléments finis associés à la triangulation T_h . On note $T_l = \{T_1, T_2, \dots, T_L\}$ les triangles de T_h , $S_I = \{S_1, T_2, \dots, S_N\}$ les sommets des triangles et $w_I = \{w_1, w_2, \dots, w_N\}$ la base de V_h définie par $\forall I, J \in \{1, 2, \dots, N\}, w_I(S_J) = \delta_{IJ}$. On cherche la solution approchée de l'équation variationnelle discrète ;

$$\boxed{\text{Trouver } u_h \in V_h \text{ tel que } \forall v_h \in V_h, a(u_h, v_h) = l(v_h)} \quad (3)$$

Qui s'écrit aussi :

$$\boxed{\text{Trouver } u_h \in V_h \text{ tel que } \forall v_h \in V_h, \int_{\Omega} (u_h v_h + \nabla u_h \cdot \nabla v_h) d\Omega = \int_{\Omega} f v_h d\Omega} \quad (4)$$

Par construction de la solution approchée (*On décompose u_h sur la base des w_I*), on a :

$$\forall (x, y) \in \bar{\Omega}^2, u_h(x, y) = \sum_{I=1}^N u_h(S_I) w_I(x, y)$$

On injecte cette expression dans l'équation variationnelle discrète et on obtient :

$$\sum_{I=1}^N u_h(S_I) \int_{\Omega} w_I v_h + \nabla w_I \cdot \nabla v_h d\Omega = \int_{\Omega} f v_h d\Omega$$

On choisit en particulier $v_h = w_J$ pour $J \in \{1, 2, \dots, N\}$, on alors :

$$\sum_{I=1}^N \left(\int_{\Omega} w_I w_J + \nabla w_I \cdot \nabla w_J d\Omega \right) u_h(S_I) = \int_{\Omega} f w_J d\Omega$$

La nouvelle formulation du problème est donc :

$$\boxed{\begin{aligned} &\text{Trouver } \vec{U} = \begin{pmatrix} U_h(S_1) \\ U_h(S_2) \\ \vdots \\ U_h(S_N) \end{pmatrix} \text{ tel que } (\mathbb{M} + \mathbb{K})\vec{U} = \vec{L} \\ &\text{où } \forall (I, J) \in \{1, 2, \dots, N\}^2, \mathbb{M}_{IJ} = \int_{\Omega} w_I w_J d\Omega, \mathbb{K}_{IJ} = \int_{\Omega} \nabla w_I \cdot \nabla w_J d\Omega, \text{ et } \vec{L}_I = \int_{\Omega} f w_I d\Omega \end{aligned}} \quad (5)$$

\mathbb{M} est symétrique :

$$\mathbb{M}_{IJ} = \int_{\Omega} w_I w_J d\Omega = \int_{\Omega} w_J w_I d\Omega = \mathbb{M}_{JI}$$

\mathbb{M} est positive :

Soit $\vec{U} \in \mathbb{R}^N$, $I \in \{1, 2, \dots, N\}$, on a :

$$(\mathbb{M}\vec{U})_I = \sum_{J=1}^N \mathbb{M}_{IJ} u_h(S_J) = \sum_{J=1}^N \left(\int_{\Omega} w_I w_J d\Omega \right) u_h(S_J) = \int_{\Omega} w_I \left(\sum_{J=1}^N w_J u_h(S_J) \right) d\Omega$$

Donc :

$$\begin{aligned} \vec{U}^T \mathbb{M} \vec{U} &= \sum_{I=1}^N u_h(S_I) (\mathbb{M}\vec{U})_I \\ &= \sum_{I=1}^N u_h(S_I) \left(\int_{\Omega} w_I \left(\sum_{J=1}^N w_J u_h(S_J) \right) d\Omega \right) \\ &= \int_{\Omega} \left(\sum_{I=1}^N w_I u_h(S_I) \right)^2 d\Omega \\ &\geq 0 \end{aligned}$$

\mathbb{M} est définie :

$$\vec{U}^T \mathbb{M} \vec{U} = 0 \Leftrightarrow \sum_{I=1}^N w_I u_h(S_I) = 0 \Leftrightarrow u_h(S_I) = 0, \forall I \in \{1, 2, \dots, N\}$$

\mathbb{K} est symétrique définie positive de manière analogue à \mathbb{M} :

Conclusion : $\boxed{\mathbb{M}, \mathbb{K} \in S_n^{++}(\mathbb{R})}$

3 Maillages

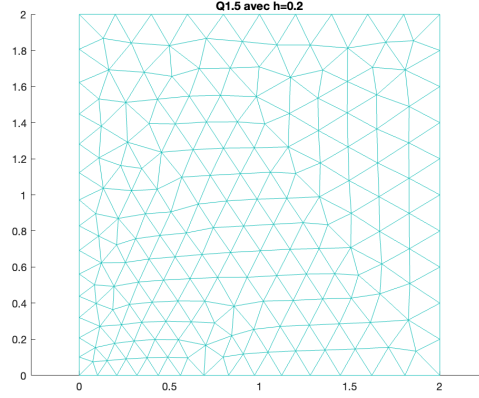


Figure 1: Maillages pour h=0.2

On a différentes structures de données lorsque l'on ouvre un maillage :

- Nbpt : nombre de sommets (entier)
- Nbtri : nombre de triangles (entier)
- Coorneu : coordonnées (x, y) des sommets (matrice réelle Nbpt x 2)
- Refneu : reference des sommets (vecteur entier Nbpt x 1)
- Numtri : liste de triangles (3 numéros de sommets - matrice entière de taille Nbtri x 3)
- Reftri : reference des triangles (matrice entière Nbtri x 1)
- Nbarettes : Nombre d'arêtes sur la frontière.
- Numarettes(Nbarettes,2) = numero des 2 Noeuds de chaque arête de la frontière
- Refarettes(Nbarettes,1) = référence de chaque arête de la frontière

4 Matrice 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 :

$$\lambda_1(x, y) = \frac{1}{D} (y_{23}(x - x_3) - x_{23}(y - y_3))$$

$$\lambda_2(x, y) = \frac{1}{D} (y_{31}(x - x_1) - x_{31}(y - y_1))$$

$$\lambda_3(x, y) = \frac{1}{D} (y_{12}(x - x_2) - x_{12}(y - y_2))$$

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 :

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

Pour calculer la matrice élémentaire \mathbb{M}^e , on doit séparer les termes de la diagonale ($i = j$) et les termes hors diagonale ($i \neq j$) :

Regardons pour \mathbb{M}_{11}^e et \mathbb{M}_{12}^e , les résultats seront les mêmes pour les autres coefficients.

$$\mathbb{M}_{11}^e = \int_{T_\ell} w_1 w_1 d\Omega = \int_{T_\ell} \lambda_1^2 d\Omega = \int_{T_\ell} \lambda_1^2 \lambda_2^0 \lambda_3^0 d\Omega = 2 \cdot \frac{2! 0! 0!}{(2+0+0+2)!} \text{aire}(T_\ell) = \frac{1}{6} \text{aire}(T_\ell) = \frac{|D|}{12}$$

$$\mathbb{M}_{12}^e = \int_{T_\ell} w_1 w_2 d\Omega = \int_{T_\ell} \lambda_1 \lambda_2 d\Omega = \int_{T_\ell} \lambda_1 \lambda_2 \lambda_3^0 d\Omega = 2 \cdot \frac{1! 1! 0!}{(1+1+0+2)!} \text{aire}(T_\ell) = \frac{1}{12} \text{aire}(T_\ell) = \frac{|D|}{24}$$

On obtient alors la matrice suivante :

$$\mathbb{M}^e = \frac{|D|}{24} \begin{pmatrix} 2 & 1 & 1 \\ 1 & 2 & 1 \\ 1 & 1 & 2 \end{pmatrix}$$

d'où le code suivant (*j'aurais pu directement utiliser le calcul précédent pour renvoyer directement la matrice élémentaire en faisant $\text{Mel} = |D|/24 * [2 \ 1 \ 1; 1 \ 2 \ 1; 1 \ 1 \ 2]$*) :

```
% preliminaires, pour faciliter la lecture:
x1 = S1(1); y1 = S1(2);
x2 = S2(1); y2 = S2(2);
x3 = S3(1); y3 = S3(2);

% D est, au signe pres, deux fois l'aire du triangle
D = ((x2-x1)*(y3-y1) - (y2-y1)*(x3-x1));
if (abs(D) <= eps)
    error('l aire d un triangle est nulle!!!');
end;

% calcul de la matrice de masse
% -----
Mel = zeros(3,3);
for i=1:3
    for j=1:3
        Mel(i,j) = D * (2 * (i == j) + 1 * (i ~= j)) / factorial (4);
    end; % j
end; % i
```

Figure 2: Code pour le calcul de la matrice élémentaire M^e

Le calcul de la matrice élémentaire \mathbb{K}^e est plus direct, en effet :

$$\nabla \lambda_1 = \frac{1}{D} \begin{pmatrix} y_2 - y_3 \\ x_3 - x_2 \end{pmatrix}$$

$$\nabla \lambda_2 = \frac{1}{D} \begin{pmatrix} y_3 - y_1 \\ x_1 - x_3 \end{pmatrix}$$

$$\nabla \lambda_3 = \frac{1}{D} \begin{pmatrix} y_1 - y_2 \\ x_2 - x_1 \end{pmatrix}$$

Or, on sait que :

$$\forall (I, J) \in \{1, 2, \dots, N\}^2, \mathbb{K}_{IJ} = \int_{\Omega} \nabla w_I \cdot \nabla w_J d\Omega = \int_{\Omega} \nabla \lambda_I \cdot \nabla \lambda_J d\Omega =$$

Pour $\Omega = T_\ell$, les λ_I sont constantes donc :

$$\forall (I, J) \in \{1, 2, \dots, N\}^2, \mathbb{K}_{IJ} = \lambda_I \lambda_J \int_{T_\ell} d\Omega = \frac{\text{norm}(i)^T \text{norm}(j)}{D^2} \frac{|D|}{2} = \frac{\text{norm}(i)^T \text{norm}(j)}{2|D|}$$

d'où le code suivant :

```

% preliminaires, pour faciliter la lecture:
x1 = S1(1); y1 = S1(2);
x2 = S2(1); y2 = S2(2);
x3 = S3(1); y3 = S3(2);

% les 3 normales a l'arete opposees (de la longueur de l'arete)
norm = zeros(3, 2);
norm(1, :) = [y2-y3, x3-x2];
norm(2, :) = [y3-y1, x1-x3];
norm(3, :) = [y1-y2, x2-x1];

% D est, au signe pres, deux fois l'aire du triangle
D = ((x2-x1)*(y3-y1) - (y2-y1)*(x3-x1));
if (abs(D) <= eps)
    error('l aire d un triangle est nulle!!!');
end;

% calcul de la matrice de raideur
% -----
Kel = zeros(3,3);
for i=1:3
    for j=1:3
        % A COMPLETER
        Kel(i,j) = (norm(i,:) * norm(j,:))/(2*D);
    end; % j
end; % i

```

Figure 3: Code pour le calcul de la matrice élémentaire K^e

5 Assemblage des matrices élémentaires

Pour assembler les matrices élémentaires, on utilise l'algorithme d'assemblage que j'ai implémenter dans Matlab.

```
% boucle sur les triangles
% -----
for l=1:Nbtri
    % Coordonnees des sommets du triangles
    % A COMPLETER
    S1 = Coorneu(Numtri(l,1),:);
    S2 = Coorneu(Numtri(l,2),:);
    S3 = Coorneu(Numtri(l,3),:);
    % calcul des matrices elementaires du triangle l

    Kel=matK_elem(S1, S2, S3);

    Mel=matM_elem(S1, S2, S3);

    % On fait l'assemblage de la matrice globale et du second membre
    for i=1:3
        I=Numtri(l,i);
        for j=1:3
            J=Numtri(l,j);
            MM(I,J)=MM(I,J)+Mel(i,j);
            KK(I,J)=KK(I,J)+Kel(i,j);
        end
    end
end % for l
```

Figure 4: Code pour l'assemblage des matrices élémentaires

6 Calcul du second membre

6.1 Calcul théorique

On peut trouver la solution \mathbb{U} en résolvant le système linéaire :

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

On sait que :

$$\forall I \in \{1, 2, \dots, N\}, L_i = l(w_i) = \int_{\Omega} f w_i d\Omega$$

On peut utiliser la méthode d'interpolation : Si $f \in C^0(\bar{\Omega})$, on l'approche par son interpolée $\pi_h f = \sum_{j=1}^{N_{\text{som}}} f(S_j) w_j$ où N_{som} est le nombre de sommets (*ie Nbpt*). Soit $i \in \{1, 2, \dots, N\}$, on a :

$$L_i \approx \int_{\Omega} \pi_h f w_i d\Omega = \sum_{j=1}^{N_{\text{som}}} f(S_j) \int_{\Omega} w_j w_i d\Omega = \sum_{j=1}^{N_{\text{som}}} f(S_j) \mathbb{M}_{ji} = (\mathbb{M}\vec{F})_i \text{ où } \vec{F}_i = f(S_i)$$

d'où $\boxed{\vec{L} = \mathbb{M}\vec{F}}$

Dans le main (*principal_neumann.m*) du code Matlab, on a alors :

```
% utiliser la routine f.m
FF = zeros(Nbpt,1);
for i=1:Nbpt
    FF(i) = f(Coorneu(i,1), Coorneu(i,2));
end
LL = MM * FF;

% inversion
% -----
UU = (MM+KK)\LL;
```

Figure 5: Code pour le calcul du second membre et résolution du système linéaire

6.2 Vérification avec $f = 1$

On obtient ce résultat pour $f = 1$ et $h = 0.2$:

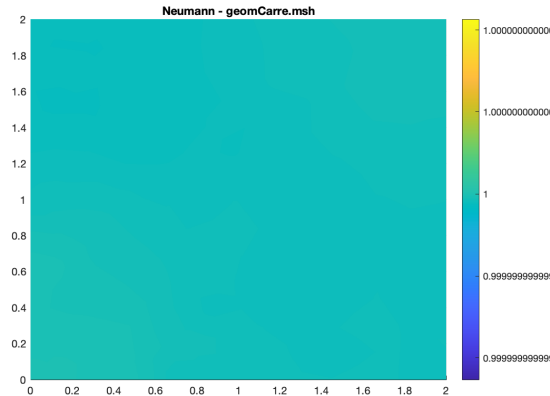


Figure 6: Solution approchée u_h pour $f = 1$ et $h = 0.2$

Il est évident que pour $f = 1$, la solution attendue de $(\mathbb{M} + \mathbb{K})\vec{U} = \vec{L}$ est $U_i = u_h(S_i) = 1 \forall i$. En affichant la solution numérique de \vec{U} dans Matlab, on obtient exactement ce qui est attendu, une solution uniforme égale à 1 partout.

7 Validation

On veut vérifier que le code calcule une solution approchée u_h correcte. Pour cela, on résout le problème (1) avec une solution connue u

$$\forall (x, y) \in \Omega, u(x, y) = \cos(\pi x) \cos(2\pi y),$$

On a :

$$\Delta u = \left(\frac{\partial^2 u}{\partial x^2} + \frac{\partial^2 u}{\partial y^2} \right) = -\pi^2 \cos(\pi x) \cos(2\pi y) - 4\pi^2 \cos(\pi x) \cos(2\pi y) = -5\pi^2 \cos(\pi x) \cos(2\pi y)$$

Comme $u - \Delta u = f$, on a :

$$\boxed{\forall (x, y) \in \Omega, f(x, y) = (1 + 5\pi^2) \cos(\pi x) \cos(2\pi y)}$$

On trouve alors :

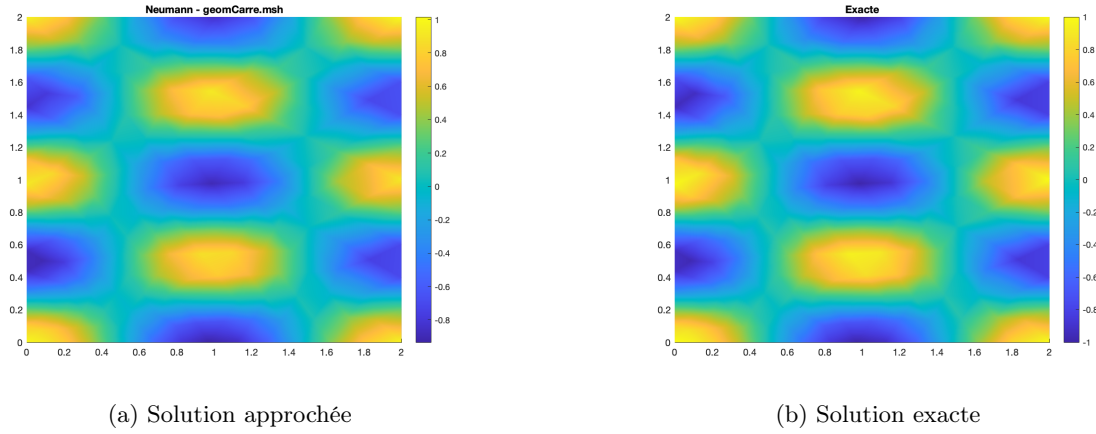


Figure 7: Résultats pour la validation pour $h = 0.2$

Les deux solutions semblent être les mêmes, pour les différencier et les comparer, on va effectuer une comparaison de la solution exacte et de la solution de Neumann en utilisant la norme L^2 de la différence des deux. On va ici encore interpolier u par $\pi_h u = \sum_{j=1}^{N_{\text{som}}} u(S_j) w_j$ où N_{som} est le nombre de sommets (ie N_{bpt})

$$\begin{aligned} \|\pi_h u - u_h\|_{L^2 \Omega} &= \left(\int_{\Omega} (\pi_h u - u_h)^2 d\Omega \right)^{1/2} \\ &= \left(\int_{\Omega} \left(\sum_{i=1}^{N_{\text{som}}} (u - u_h)(S_i) w_i \right) \left(\sum_{j=1}^{N_{\text{som}}} (u - u_h)(S_j) w_j \right) d\Omega \right)^{1/2} \\ &= \left(\sum_{i=1}^{N_{\text{som}}} \sum_{j=1}^{N_{\text{som}}} (u - u_h)(S_i) \left(\int_{\Omega} w_i w_j d\Omega \right) (u - u_h)(S_j) \right)^{1/2} \end{aligned}$$

On a alors :

$$\boxed{\|u - u_h\|_{L^2 \Omega} \approx \|\pi_h u - u_h\|_{L^2 \Omega} = \left((U_{\text{exact}}^{\vec{}} - \vec{U})^T \mathbb{M} (U_{\text{exact}}^{\vec{}} - \vec{U}) \right)^{1/2}}$$

De même :

$$\|u\|_{L^2 \Omega} \approx \|\pi_h u\|_{L^2 \Omega} = \left((U_{\text{exact}}^{\vec{}})^T \mathbb{M} (U_{\text{exact}}^{\vec{}}) \right)^{1/2}$$

On peut alors calculer l'erreur relative :

$$\frac{\|u - u_h\|_{L^2\Omega}}{\|u\|_{L^2\Omega}} = \frac{\left((U_{exact}^{\vec{}} - \vec{U})^T \mathbb{M}(U_{exact}^{\vec{}} - \vec{U})\right)^{1/2}}{\left((U_{exact}^{\vec{}})^T \mathbb{M}(U_{exact}^{\vec{}})\right)^{1/2}}$$

De manière analogue, on obtient pour la semi-norme H^1 :

$$|u - u_h|_{H^1\Omega} \approx |\pi_h u - u_h|_{H^1\Omega} = \left((U_{exact}^{\vec{}} - \vec{U})^T \mathbb{K}(U_{exact}^{\vec{}} - \vec{U})\right)^{1/2}$$

Et l'erreur relative :

$$\frac{|u - u_h|_{H^1\Omega}}{|u|_{H^1\Omega}} = \frac{\left((U_{exact}^{\vec{}} - \vec{U})^T \mathbb{K}(U_{exact}^{\vec{}} - \vec{U})\right)^{1/2}}{\left((U_{exact}^{\vec{}})^T \mathbb{K}(U_{exact}^{\vec{}})\right)^{1/2}}$$

On implémente cela dans le code Matlab :

```
validation = 'oui';
% validation
% -----
if strcmp(validation, 'oui')
    UU_exact = cos(pi*Coorneu(:,1)).*cos(2*pi*Coorneu(:,2));
    affiche(UU_exact, Numtri, Coorneu, "Exacte");
    % Calcul de l'erreur L2
    erreur_L2 = sqrt((UU_exact-UU)'*MM*(UU_exact-UU))
    % Calcul de l'erreur relative L2
    erreur_relative_L2 = sqrt((UU_exact-UU)'*MM*(UU_exact-UU)) / sqrt((UU_exact)'*MM*(UU_exact))
    % Calcul de l'erreur semi H1
    erreur_semi_H1 = sqrt((UU_exact-UU)'*KK*(UU_exact-UU))
    % Calcul de l'erreur relative semi H1
    erreur_relative_semi_H1 = sqrt((UU_exact-UU)'*KK*(UU_exact-UU)) / sqrt((UU_exact)'*KK*(UU_exact))
end
```

Figure 8: Code pour le calcul des erreurs

On peut maintenant calculer ces erreurs pour différentes valeurs de h que l'on choisira de telle manière que $h_{i+1} = \frac{h_i}{2}$ afin de pouvoir effectuer une échelle $\log - \log$

h	$\ u - u_h\ _{L^2(\Omega)}$	$\frac{\ u - u_h\ _{L^2(\Omega)}}{\ u\ _{L^2(\Omega)}}$	$ u - u_h _{H^1(\Omega)}$	$\frac{ u - u_h _{H^1(\Omega)}}{ u _{H^1(\Omega)}}$
0.2	0.1021	0.1117	0.6088	0.0906
0.1	0.0261	0.0267	0.1784	0.0257
0.05	0.0065	0.0065	0.0570	0.0081
0.025	0.0016	0.0016	0.0178	0.0025

Table 1: Tableau des résultats des erreurs

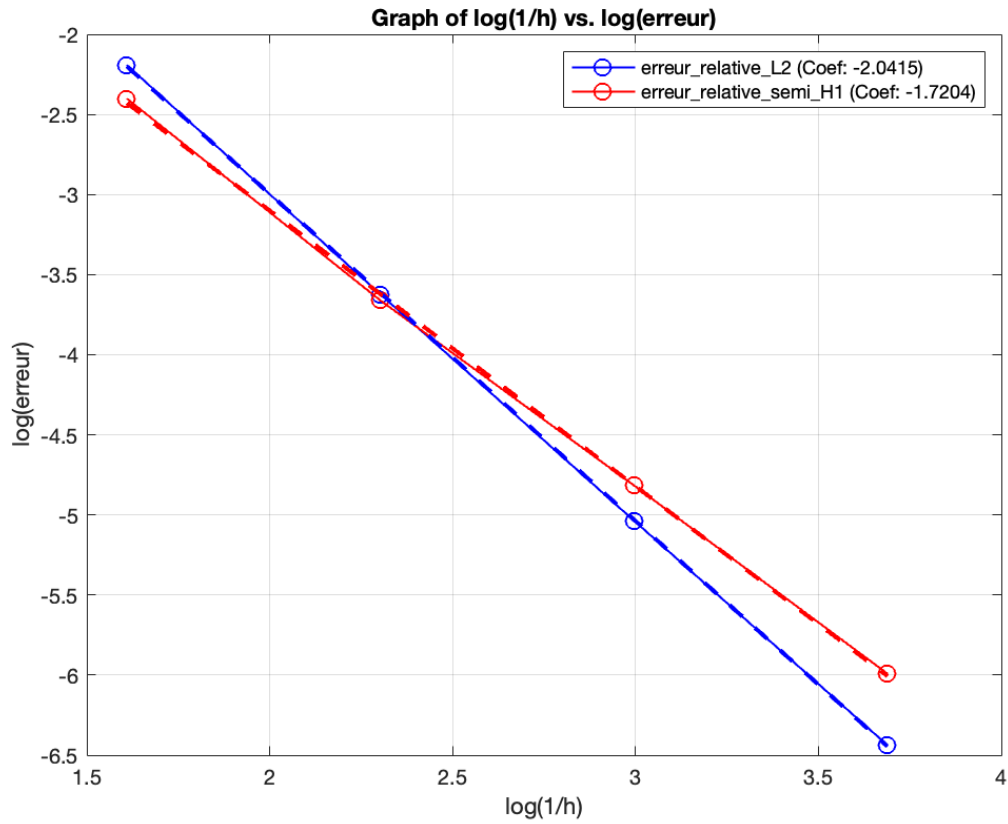


Figure 9: Graphique des erreurs

On peut observer que plus on réduit le pas de maillage, plus l'erreur est petite. C'est un résultat que l'on attend, un maillage plus raffiné donne des résultats plus précis, mais il nécessite également un plus grand temps de calculs informatiques.

8 Résolution d'un problème similaire

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

$$\begin{cases} -\Delta u = f & \text{dans } \Omega, \\ \frac{\partial u}{\partial n} = 0 & \text{sur } \partial\Omega. \end{cases} \quad (6)$$

revient à résoudre le problème suivant en formulation variationnelle en appliquant la même méthode que précédemment :

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

$$\forall v \in H^1(\Omega), \int_{\Omega} \nabla u \cdot \nabla v \, d\Omega = \int_{\Omega} f v \, d\Omega$$

(7)

Il faut cependant faire attention au théorème de Lax-Milgram, en effet, la forme bilinéaire est toujours continue (*la constante vaut 1 au lieu de 2 maintenant*) mais elle n'est plus forcément coercive, il faut le vérifier.

On rappelle la définition de a :

$$a : \begin{array}{ccc} H^1(\Omega) \times H^1(\Omega) & \longrightarrow & \mathbb{R} \\ (u, v) & \longmapsto & \int_{\Omega} \nabla u \cdot \nabla v \, d\Omega \end{array}$$

On a pour tout $u \in H^1(\Omega)$:

$$|a(u, u)| = \|\nabla u\|_{L^2(\Omega)}^2$$

En se plaçant dans $H_0^1(\Omega)$, on a par l'inégalité de Poincaré :

$$\|u\|_{L^2(\Omega)}^2 \leq C_p \|\nabla u\|_{L^2(\Omega)}^2$$

Or comme :

$$\|u\|_{H^1(\Omega)}^2 = \|u\|_{L^2(\Omega)}^2 + \|\nabla u\|_{L^2(\Omega)}^2$$

On a alors :

$$\|u\|_{H^1(\Omega)}^2 \leq (1 + C_p) \|\nabla u\|_{L^2(\Omega)}^2$$

On injecte cela dans l'expression de a :

$$|a(u, u)| = \|\nabla u\|_{L^2(\Omega)}^2 \geq \frac{1}{1 + C_p} \|u\|_{H^1(\Omega)}^2, \quad C_p > 0$$

Donc a est coercive sur $H_0^1(\Omega)$, et le problème est bien posé.

En discrétisant le problème, on obtient la formulation suivante :

Trouver $\vec{U} = \begin{pmatrix} U_h(S_1) \\ U_h(S_2) \\ \vdots \\ U_h(S_N) \end{pmatrix}$ tel que $\mathbb{K}\vec{U} = \vec{L}$

où $\forall (I, J) \in \{1, 2, \dots, N\}^2$, $\mathbb{K}_{IJ} = \int_{\Omega} \nabla w_I \cdot \nabla w_J \, d\Omega$, et $\vec{L}_I = \int_{\Omega} f w_I \, d\Omega$

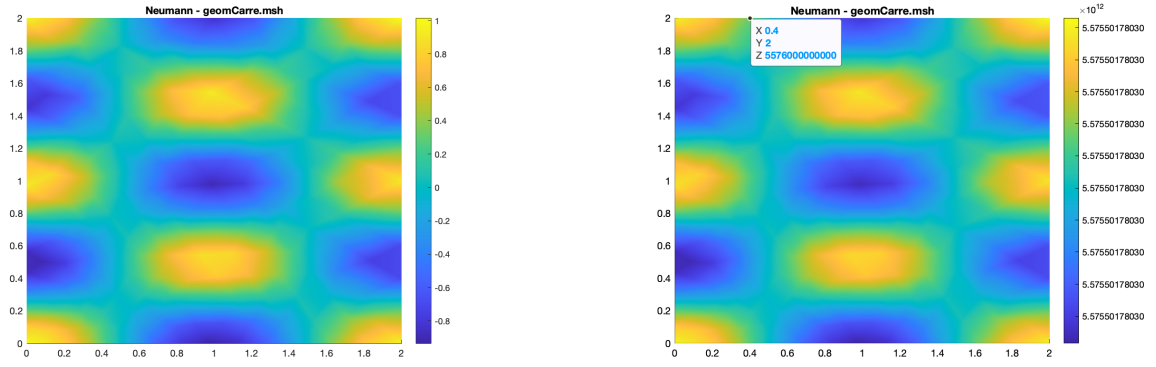
(8)

Cela revient à considérer $\mathbb{M} = \mathbf{0}_{N \times N}$ dans le membre de gauche de l'équation, le calcul du membre de droite reste lui inchangé.

```
% inversion
% -----
UU = KK\LL;
```

Figure 10: Changement dans le code pour le problème similaire

Sur l'exemple suivant : f comme dans la partie validation et $h = 0.2$, on obtient la solution suivante :



(a) solution pour $u - \Delta u = f$

(b) solution pour $-\Delta u = f$

Figure 11: Comparaison des solutions pour les deux problèmes

Les solutions obtenues pour les deux problèmes sont semblables mais la différence majeure réside dans les échelles. Le terme en u du problème initial semble contrôler l'échelle de la solution, ce qui n'est pas le cas pour le second problème (*on trouve aussi le même résultat pour la solution $f = 1$*).