

# Compte Rendu TP MEF

Alexandre Prampart / Matthieu Petit

Mai 2023

# Table des matières

1	Instructions pour le code . . . . .	2
1.1	Contenu . . . . .	2
1.2	Bash . . . . .	2
1.3	Codes Python . . . . .	3
2	Domaine 1 . . . . .	4
2.1	Problème 1 . . . . .	4
2.2	Problème 2 . . . . .	7
2.3	Problème 3 . . . . .	9
3	Domaine 2 . . . . .	11
3.1	Problème 1 . . . . .	12
3.2	Problème 2 . . . . .	14
3.3	Problème 3 . . . . .	16
4	Quelques commentaires et analyses . . . . .	18

# 1 Instructions pour le code

## 1.1 Contenu

Le code est compartimenté en plusieurs dossiers :

- FonctionC
- FonctionsFortran
- Header

Le dossier 'FonctionC' comprend le sous-dossier 'Main' avec les différents main utiles (en l'occurrence 2, voir la sous-section Bash 1.2), les différentes fonctions 'hautes' comme le calcul de la solution exacte et le menu, ces fonctions sont donc à changer si l'on veut résoudre d'autres problèmes.

Les fonctions de calcul sont quant à elles dans le sous-dossier 'FctsCalcul' et celles pour l'assemblage sont dans le sous-dossier 'FctsAssemblage'.

Un autre fichier important se nomme 'fdp.c', il contient la définition des fonctions inhérentes à l'opérateur différentiel :  $a_{ij}$ ,  $b_n$ ,  $f_n$  etc. Il sera donc également important de changer ces fonctions là si l'on s'intéresse à d'autres problèmes.

Le code va utiliser des fichiers de maillage et de numéros de références des arêtes et du domaine, ces informations sont stockées dans les fichiers :

- FicMaillage
- NumRef

Les fichiers dans le répertoire 'NumRef' sont définis pour chaque problème et chaque domaine. Le dossier 'FicMaillaTest' contient les maillages utilisés pour les tests des TPs précédents.

Enfin, nous trouvons également le répertoire 'Images' contenant les courbes d'erreur tracées à l'aide de nos résultats ainsi que la visualisation de la solution calculée pour quelques problème et domaine. Toutes ces figures sont produites par les codes python 'graphe.py' et 'sol3D.py' contenus dans le dossier 'Python'.

## 1.2 Bash

Deux scripts bash ont été écrit, ils permettent, dans un premier temps, de compiler le code et de faire l'édition de lien entre toutes les fonctions contenues dans chaque dossier, qu'elles soient des fonctions Fortran ou C. Notamment en sélectionnant un main différent en fonction de si l'on veut un menu ou bien un boucle sur chaque problème.

Dans un second temps le bash nommé 'bashMenu.sh' exécute (si la compilation et l'édition de lien s'est bien passé) le code d'éléments finis en affichant en premier un menu, permettant à

l'utilisateur de sélectionner un domaine, un problème, un maillage et une condition d'affichage. Quant au bash 'bashBoucle.sh', il exécute plusieurs fois le codes éléments finis, pour chaque problème, chaque domaine, et chaque maillage (taille du maillage, triangle, quadrangle). Il imprime ensuite les résultats dans plusieurs fichiers stockés dans le sous-dossier 'FichiersResultats' qui seront ensuite utilisé afin de calculer les courbes d'erreurs l'affichage de solutions calculées.

On peut utiliser la ligne de commande '**./bashMenu.sh**' par exemple. Il est possible d'avoir à configurer l'accès des fichiers avec la commande '**chmod 755 bashMenu.sh**' avant.

### 1.3 Codes Python

Le sous-dossier 'Python' comporte deux fichier '.py'.

Le fichier 'graphe.py' récupère les fichiers résultats 'fort.\*' stockés dans le sous-dossier 'Fichiers-Resultats'. Il calcule ensuite les courbes d'erreur pour chaque domaine et chaque problème et imprime des courbes dans le sous dossier 'Images/Courbes-Erreurs'.

De même, le code 'sol3D.py' va récupérer les fichiers résultats 'solCalc\*' stockés dans le sous-dossier 'FichiersResultats' et imprime dans le sous dossier 'Image/Figures-Calculees' un graphe 3D de la solution calculée pour différents angles, et ça pour chaque domaine et chaque problème.

On peut utiliser la ligne de commande '**python3 graphe.py**' ou bien '**python3 sol3D.py**' par exemple afin d'exécuter ces codes python.

**Note :** N'hésitez pas à revenir vers nous si un problème apparaît ou si vous avez une question sur une partie du code.

On rappelle l'équation de la formulation variationnelle que nous avons programmée :  
 Trouver  $u \in W$  tel que  $u = u_D$  sur  $\Gamma_D$  et

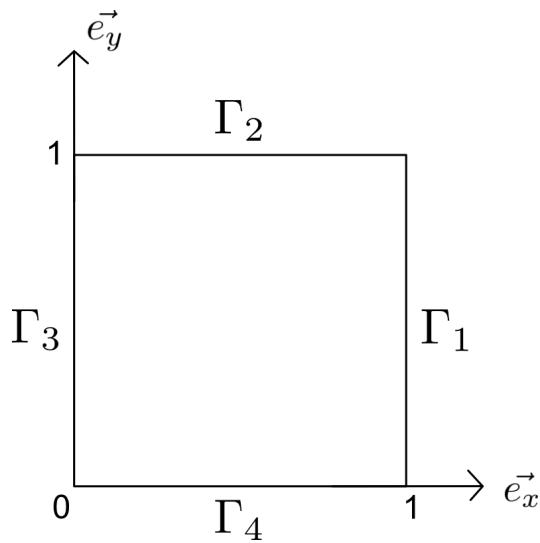
$$a(u, v) = \ell(v) \quad \forall v \in V$$

Avec :

$$a(u, v) = \int_{\Omega} \left( \sum_{\alpha, \beta=1}^2 a_{\alpha\beta}(x) \frac{\partial u}{\partial x_{\alpha}}(x) \frac{\partial v}{\partial x_{\beta}}(x) + a_{00}(x) u(x)v(x) \right) dx + \int_{\Gamma_N} b_n(\gamma) u(\gamma)v(\gamma) d\gamma \quad (1)$$

$$\ell(v) = \int_{\Omega} f_{\Omega}(x)v(x) dx + \int_{\Gamma_N} f_N(\gamma)v(\gamma) d\gamma \quad (2)$$

## 2 Domaine 1



### 2.1 Problème 1

Le problème à résoudre est le suivant :

$$-\Delta u = f_{\Omega}$$

La solution exacte est donnée pour  $u = 16xy(1-x)(1-y)$ , ainsi on peut en déduire que :

$$f_{\Omega} = 32(x(1-x) + y(1-y))$$

On impose une condition de Dirichlet homogène sur les 4 côtés, ainsi on a :

$$\Gamma_D = \Gamma_1 \cup \Gamma_2 \cup \Gamma_3 \cup \Gamma_4$$

$$\Gamma_N = \emptyset$$

On doit également définir toutes les fonctions de la formulation variationnelle comme suit :

$$a_{\alpha\beta} = \delta_{\alpha\beta}$$

$$a_{00} = 0$$

$$b_N = 0$$

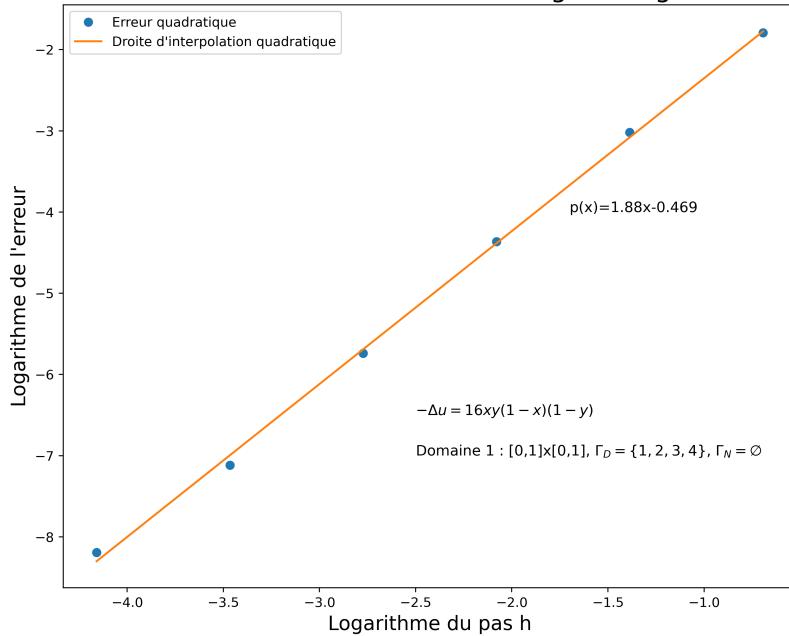
$$f_N = 0$$

Après avoir exécuté le programme des éléments finis pour des maillages de plus en plus fins, on peut tracer une courbe qui nous montre l'erreur comparée à la solution exacte en fonction de la finesse du maillage.

Dans un premier temps on utilise des maillages constitués de triangles, on obtient la courbe suivante :

### Etude de l'erreur en fonction du pas de discréétisation

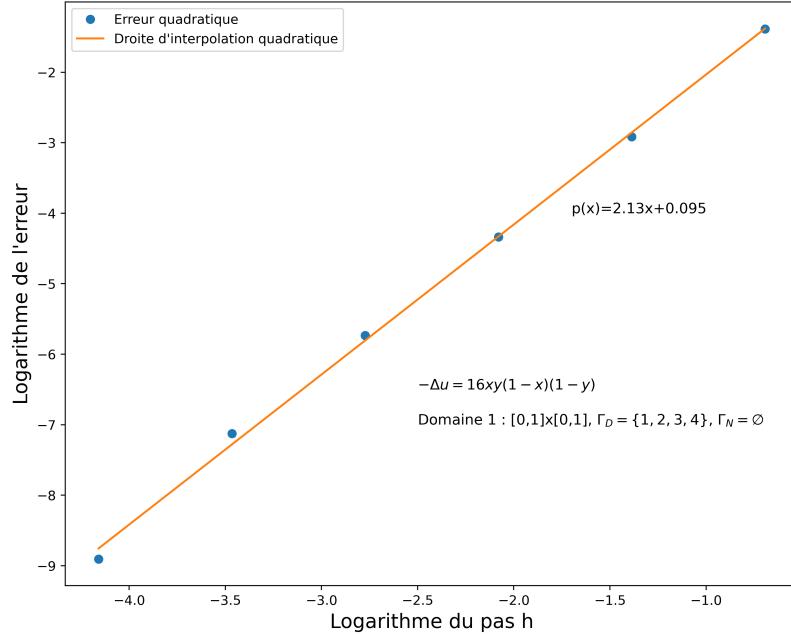
Domaine 1 Problème 1 Maillage Triangle



Dans un second temps, pour le même problème on utilise un maillage de quadrangle, on obtient alors la courbe suivante :

## Etude de l'erreur en fonction du pas de discréétisation

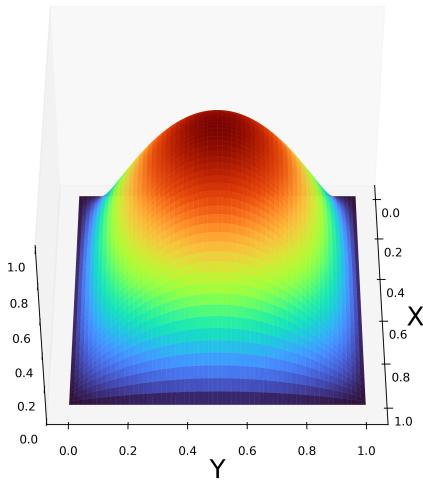
Domaine 1 Problème 1 Maillage Quadrangle



On peut observer ici que l'interpolation des différents points nous donne un meilleur coefficient sur le maillage de type quadrangle (2.13) que de type triangle (1.88).

On peut maintenant afficher la solution calculée en 3D pour visualiser la solution.

Solution pour le Domaine 1 Problème 1



Solution pour le Domaine 1 Problème 1

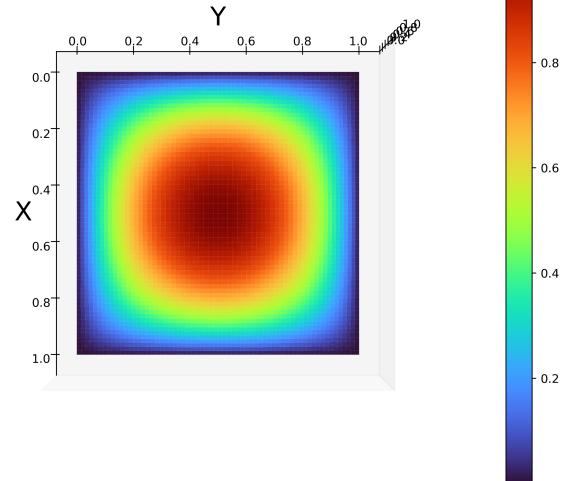


FIGURE 1 – Solution calculée vue sous différents angles Domaine 1 Problème 1

## 2.2 Problème 2

Le problème à résoudre est le suivant :

$$-\Delta u = f_\Omega$$

La solution exacte est donnée pour  $u = \sin(\pi x) \sin(\pi y)$ , ainsi on peut en déduire que :

$$f_\Omega = 2\pi^2 \sin(\pi x) \sin(\pi y)$$

On impose également une condition de Dirichlet homogène sur les 4 côtés, ainsi on :

$$\Gamma_D = \Gamma_1 \cup \Gamma_2 \cup \Gamma_3 \cup \Gamma_4$$

$$\Gamma_N = \emptyset$$

Ici encore on doit également définir toutes les fonctions de la formulation variationnelles qui sont les mêmes que dans le problème 1 :

$$a_{\alpha\beta} = \delta_{\alpha\beta}$$

$$a_{00} = 0$$

$$b_N = 0$$

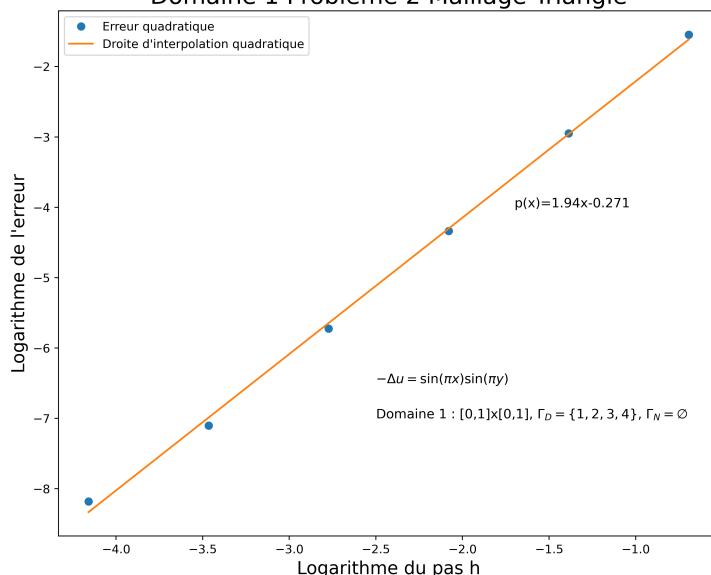
$$f_N = 0$$

Après avoir exécuté le programme des éléments finis pour des maillages de plus en plus fin, on peut tracer une courbe qui nous montre l'erreur comparée à la solution exacte en fonction de la finesse du maillage.

Dans un premier temps on utilise des maillages constitués de triangles, on obtient la courbe suivante :

**Etude de l'erreur en fonction du pas de discréétisation**

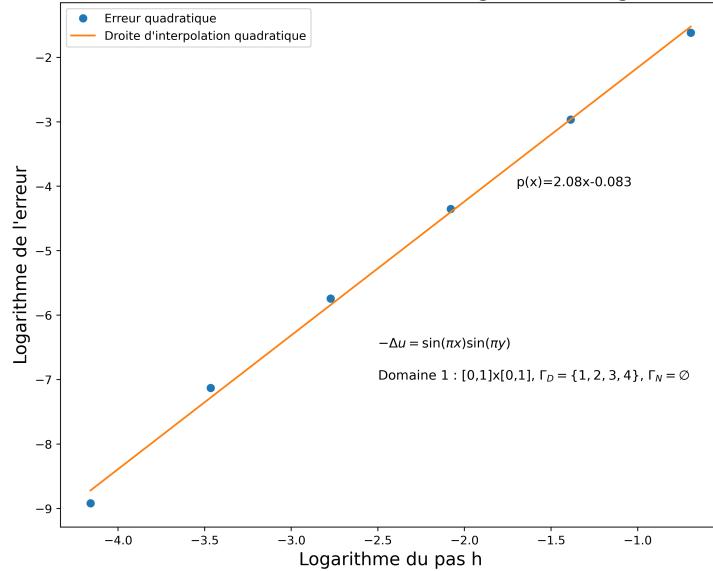
Domaine 1 Problème 2 Maillage Triangle



Dans un second temps, pour le même problème on utilise un maillage de quadrangle, on obtient alors la courbe suivante :

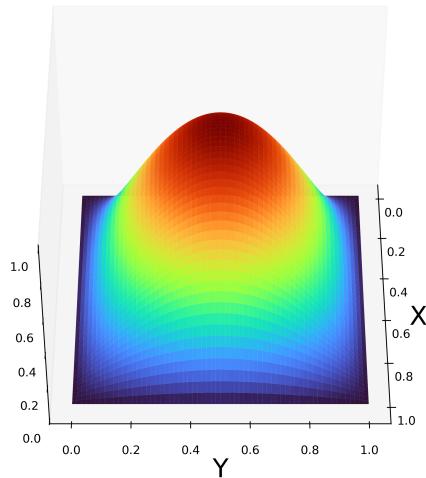
### Etude de l'erreur en fonction du pas de discréétisation

#### Domaine 1 Problème 2 Maillage Quadrangle



On peut maintenant afficher la solution calculée en 3D pour visualiser la solution.

Solution pour le Domaine 1 Problème 2



Solution pour le Domaine 1 Problème 2

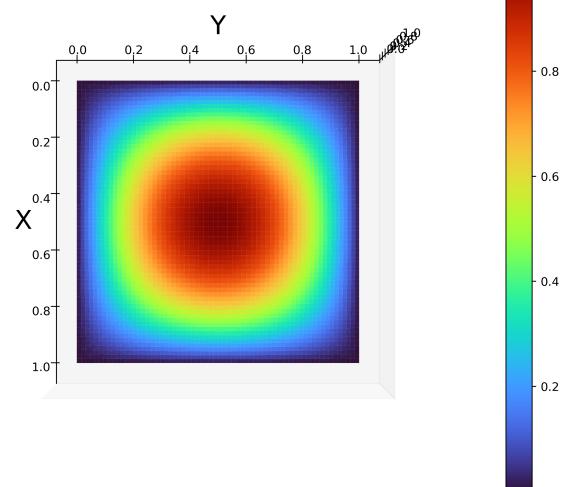


FIGURE 2 – Solution calculée vue sous différents angles Domaine 1 Problème 2

### 2.3 Problème 3

Le problème à résoudre est le suivant :

$$\begin{cases} -\Delta u + u = f_\Omega \\ \Gamma_D = \emptyset \\ \Gamma_N = \Gamma_1 \cup \Gamma_2 \cup \Gamma_3 \cup \Gamma_4 \end{cases} \quad (3)$$

La solution exacte est donnée pour  $u = \cos(\pi x) \cos(\pi y)$ , ainsi on peut en déduire que :

$$\begin{aligned} \Delta u(x, y) &= -2\pi^2 \cos(\pi x) \cos(\pi y) \\ \Rightarrow f_\Omega &= (1 + 2\pi^2) \cos(\pi x) \cos(\pi y) \end{aligned}$$

On impose ici une condition de Neumann homogène sur les 4 côtés, ainsi on a :

$$\begin{aligned} \Gamma_D &= \emptyset \\ \Gamma_N &= \Gamma_1 \cup \Gamma_2 \cup \Gamma_3 \cup \Gamma_4 \end{aligned}$$

On doit donc également déduire  $f_N$ .

On a  $\vec{n}|_{\Gamma_1} = \vec{e}_x$ ,  $\vec{n}|_{\Gamma_2} = \vec{e}_y$ ,  $\vec{n}|_{\Gamma_3} = -\vec{e}_x$  et  $\vec{n}|_{\Gamma_4} = -\vec{e}_y$  et également :

$\nabla u(x, y) = \left( -\pi \sin(\pi x) \cos(\pi y), -\pi \cos(\pi x) \sin(\pi y) \right)^T$  on peut donc en déduire que :

$$\begin{aligned} \frac{\partial u}{\partial \vec{n}|_{\Gamma_1}} &= \nabla u \cdot \vec{e}_x = -\pi \sin(\pi x) \cos(\pi y) \\ \frac{\partial u}{\partial \vec{n}|_{\Gamma_2}} &= \nabla u \cdot \vec{e}_y = -\pi \cos(\pi x) \sin(\pi y) \\ \frac{\partial u}{\partial \vec{n}|_{\Gamma_3}} &= \nabla u \cdot (-\vec{e}_x) = \pi \sin(\pi x) \cos(\pi y) \\ \frac{\partial u}{\partial \vec{n}|_{\Gamma_4}} &= \nabla u \cdot (-\vec{e}_y) = \pi \cos(\pi x) \sin(\pi y) \end{aligned}$$

Ainsi on peut définir la fonction  $f_N$  comme suit :

$$f_N = \begin{cases} -\pi \sin(\pi x) \cos(\pi y) & \text{sur } \Gamma_1 \\ -\pi \cos(\pi x) \sin(\pi y) & \text{sur } \Gamma_2 \\ \pi \sin(\pi x) \cos(\pi y) & \text{sur } \Gamma_3 \\ \pi \cos(\pi x) \sin(\pi y) & \text{sur } \Gamma_4 \end{cases}$$

Il faut alors faire attention à ce que les conditions aux bords soit placées sur les bonnes arêtes, il faut donc corréler la définition de la fonction  $f_N$  avec le maillage que l'on prend en entrée.

On doit également définir toutes les fonctions de la formulation variationnelles comme suit :

$$a_{\alpha\beta} = \delta_{\alpha\beta}$$

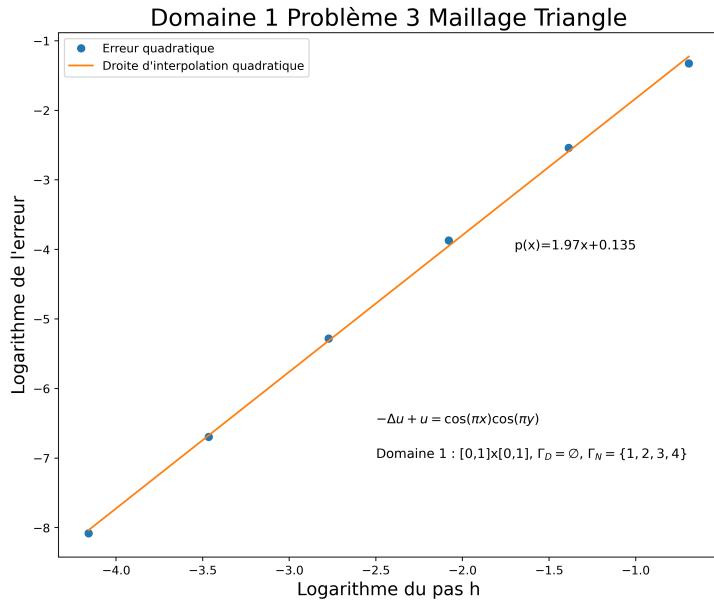
$$a_{00} = 1$$

$$b_N = 0$$

Après avoir exécuté le programme des éléments finis pour des maillages de plus en plus fin, on peut tracer une courbe qui nous montre l'erreur comparée à la solution exacte en fonction de la finesse du maillage.

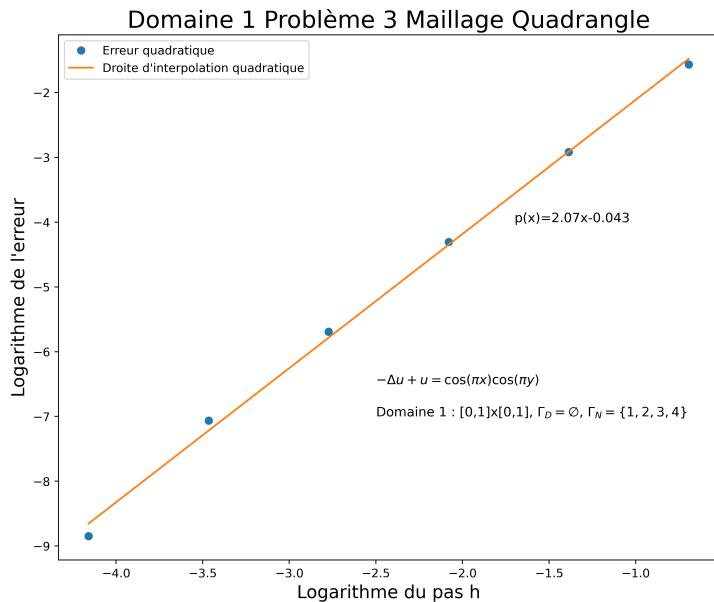
Dans un premier temps on utilise des maillages constitués de triangles, on obtient la courbe suivante :

### Etude de l'erreur en fonction du pas de discréétisation



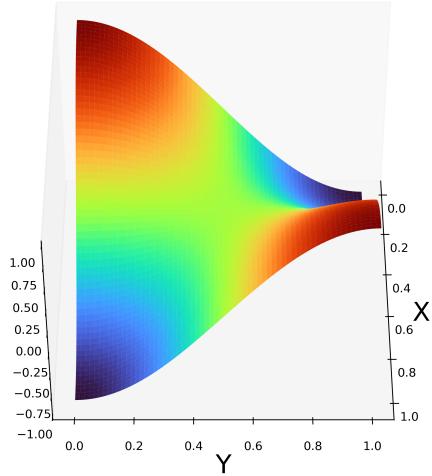
Dans un second temps, pour le même problème on utilise un maillage de quadrangle, on obtient alors la courbe suivante :

### Etude de l'erreur en fonction du pas de discréétisation



On peut maintenant afficher la solution calculée en 3D pour visualiser la solution.

Solution pour le Domaine 1 Problème 3



Solution pour le Domaine 1 Problème 3

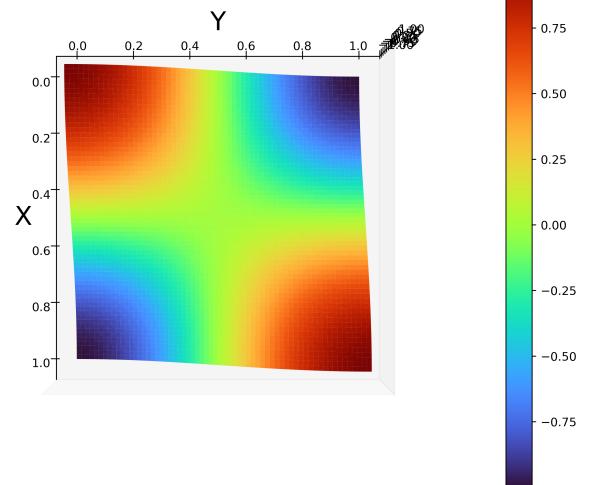
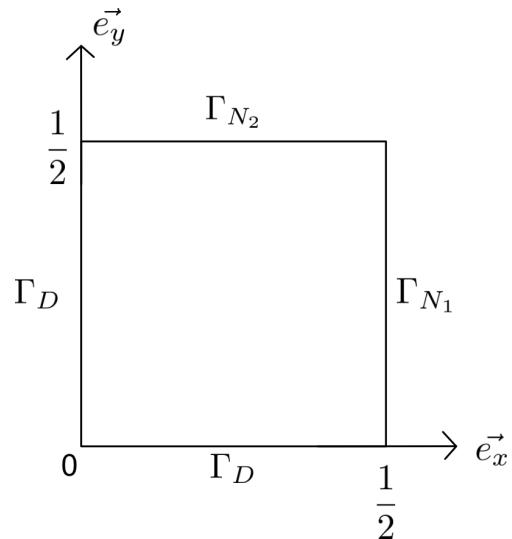


FIGURE 3 – Solution calculée vue sous différents angles Domaine 1 Problème 3

### 3 Domaine 2



Pour le domaine 2 on a :

$$\begin{aligned}\vec{n}|_{\Gamma_{N_1}} &= \vec{e}_x \\ \vec{n}|_{\Gamma_{N_2}} &= \vec{e}_y\end{aligned}$$

### 3.1 Problème 1

Pour la solution  $u = 16xy(1-x)(1-y)$  :

On a dans un premier temps :

$$\nabla u(x, y) = \left( 16x(1-x), 16y(1-y) \right)^T$$

On peut donc calculer  $f_N$  :

$$f_N = \begin{cases} 16x(1-x) & \text{sur } \Gamma_1 \\ 16y(1-y) & \text{sur } \Gamma_2 \end{cases}$$

On a maintenant la définition des autres fonctions de la formulation variationnelle :

$$a_{\alpha\beta} = \delta_{\alpha\beta}$$

$$a_{00} = 0$$

$$b_N = 0$$

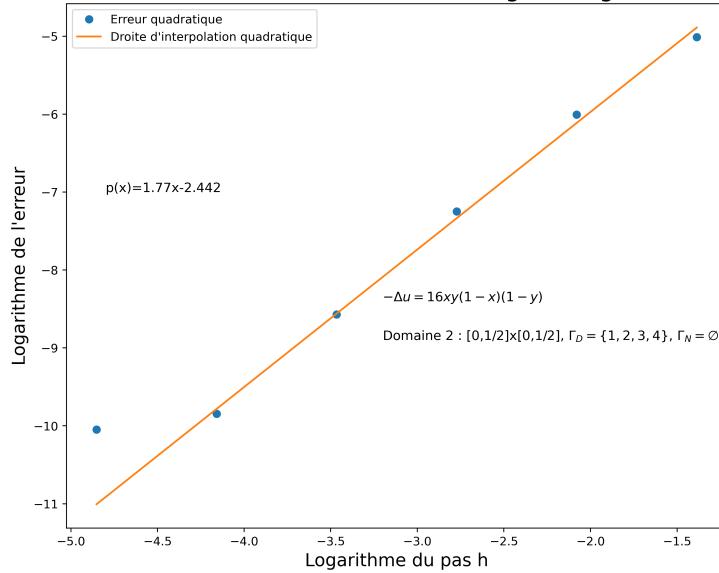
$$f_\Omega = 32(x(1-x) + y(1-y))$$

$$u_D = 16xy(1-x)(1-y)$$

On peut maintenant exécuter notre programme pour chaque maillage. Dans un premier temps on utilise des maillages constitués de triangles, on obtient la courbe suivante :

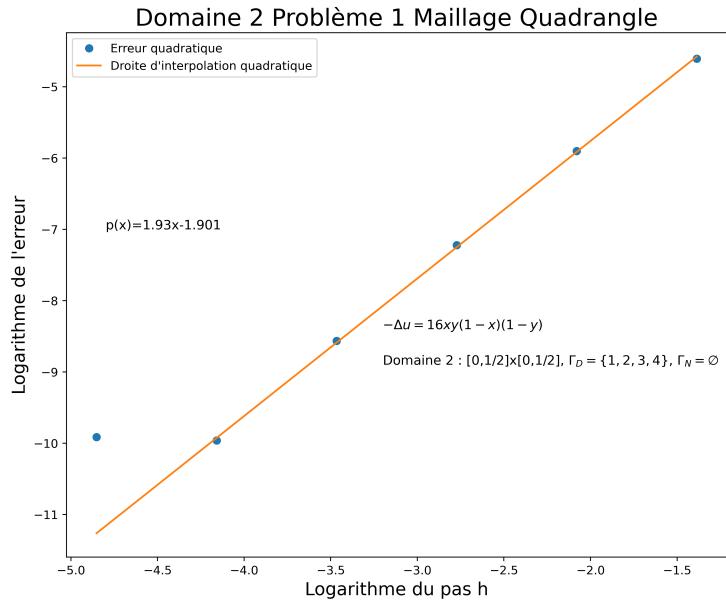
Etude de l'erreur en fonction du pas de discréétisation

Domaine 2 Problème 1 Maillage Triangle



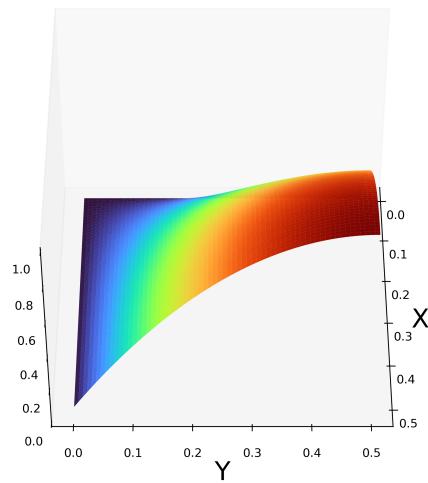
Dans un second temps, pour le même problème on utilise un maillage de quadrangle, on obtient alors la courbe suivante :

## Etude de l'erreur en fonction du pas de discréétisation



On peut maintenant afficher la solution calculée en 3D pour visualiser la solution.

Solution pour le Domaine 2 Problème 1



Solution pour le Domaine 2 Problème 1

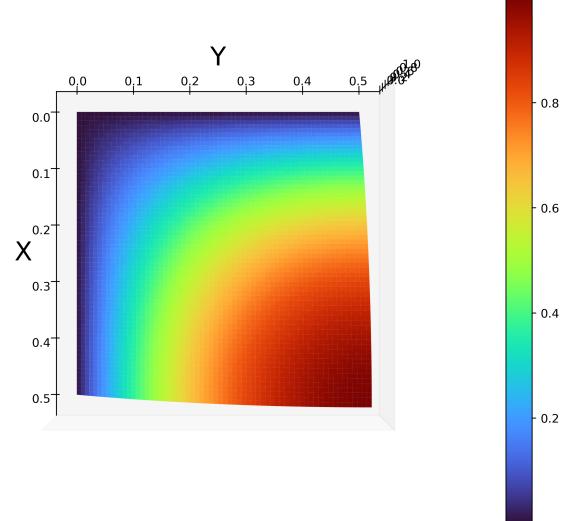


FIGURE 4 – Solution calculée vue sous différents angles Domaine 2 Problème 1

### 3.2 Problème 2

Pour la solution  $u = \sin(\pi x) \sin(\pi y)$  :

$$\nabla u(x, y) = \left( \pi \cos(\pi x) \sin(\pi y), \pi \sin(\pi x) \cos(\pi y) \right)^T$$

On peut donc calculer  $f_N$  :

$$f_N = \begin{cases} \pi \cos(\pi x) \sin(\pi y) & \text{sur } \Gamma_1 \\ \pi \sin(\pi x) \cos(\pi y) & \text{sur } \Gamma_2 \end{cases}$$

Ici encore on a toujours la définition des autres fonctions :

$$a_{\alpha\beta} = \delta_{\alpha\beta}$$

$$a_{00} = 0$$

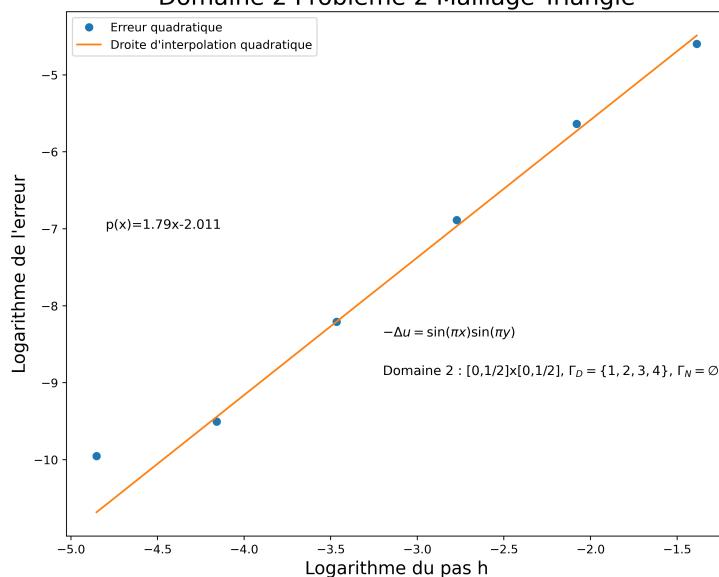
$$b_N = 0$$

$$u_D = \sin(\pi x) \sin(\pi y)$$

$$f_\Omega = 2\pi^2 \sin(\pi x) \sin(\pi y)$$

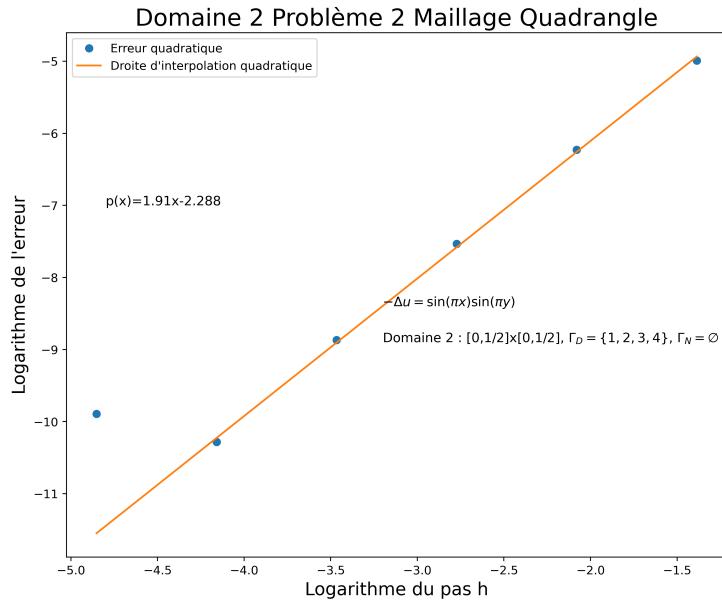
On peut maintenant exécuter notre programme pour chaque maillage. Dans un premier temps on utilise des maillages constitué de triangles, on obtient la courbe suivante :

**Etude de l'erreur en fonction du pas de discréétisation**  
**Domaine 2 Problème 2 Maillage Triangle**



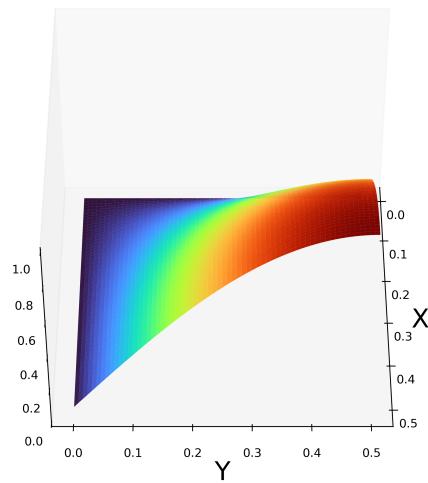
Dans un second temps, pour le même problème on utilise un maillage de quadrangle, on obtient alors la courbe suivante :

## Etude de l'erreur en fonction du pas de discréétisation



On peut maintenant afficher la solution calculée en 3D pour visualiser la solution.

Solution pour le Domaine 2 Problème 2



Solution pour le Domaine 2 Problème 2

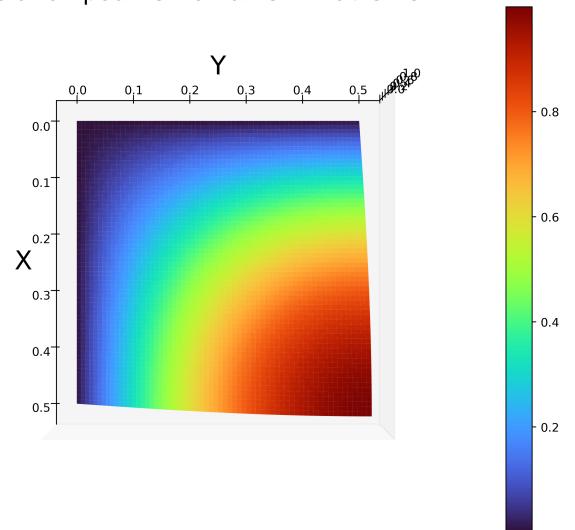


FIGURE 5 – Solution calculée vue sous différents angles Domaine 2 Problème 2

### 3.3 Problème 3

Pour la solution  $u = \cos(\pi x) \cos(\pi y)$  :

$$\nabla u(x, y) = \left( -\pi \sin(\pi x) \cos(\pi y), -\pi \cos(\pi x) \sin(\pi y) \right)^T$$

On peut donc calculer  $f_N$  :

$$f_N = \begin{cases} -\pi \sin(\pi x) \cos(\pi y) & \text{sur } \Gamma_1 \\ -\pi \cos(\pi x) \sin(\pi y) & \text{sur } \Gamma_2 \end{cases}$$

Ici encore on a toujours la définition des autres fonctions :

$$a_{\alpha\beta} = \delta_{\alpha\beta}$$

$$a_{00} = 1$$

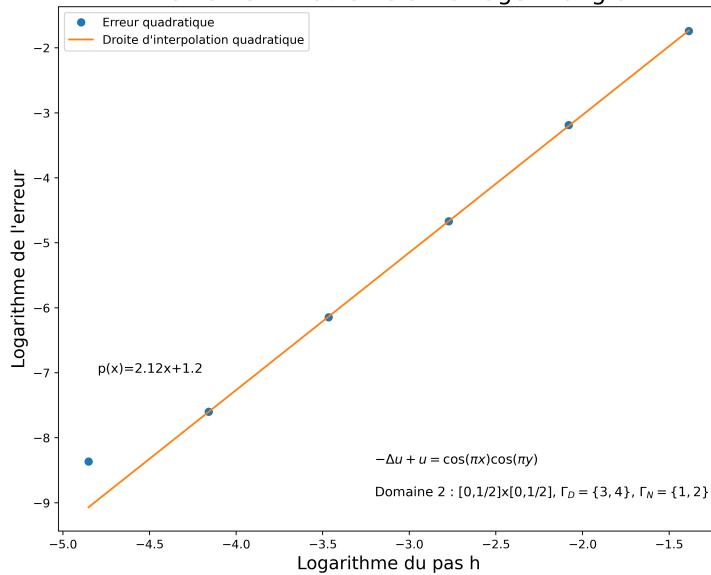
$$b_N = 0$$

$$u_D = \cos(\pi x) \cos(\pi y)$$

$$f_\Omega = (1 + 2\pi^2) \cos(\pi x) \cos(\pi y)$$

On peut maintenant exécuter notre programme pour chaque maillage. Dans un premier temps on utilise des maillages constitués de triangles, on obtient la courbe suivante :

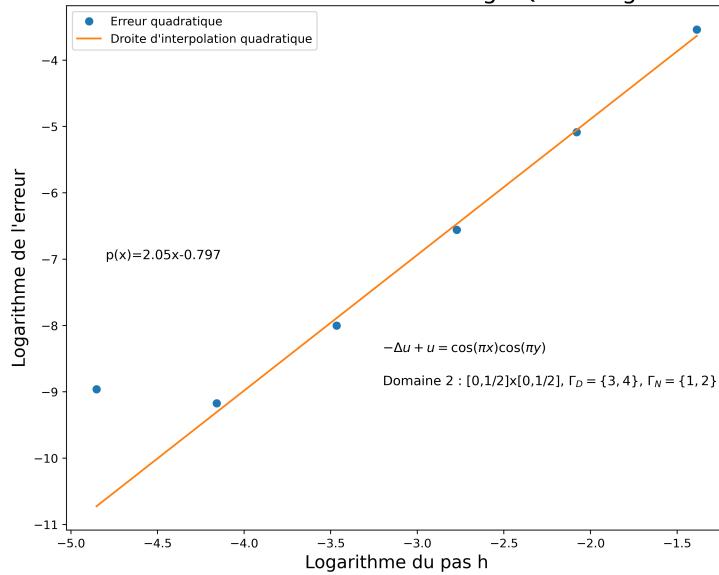
**Etude de l'erreur en fonction du pas de discréétisation**  
**Domaine 2 Problème 3 Maillage Triangle**



Dans un second temps, pour le même problème on utilise un maillage de quadrangle, on obtient alors la courbe suivante :

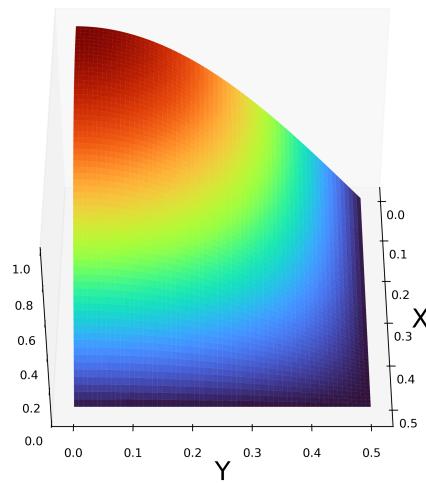
## Etude de l'erreur en fonction du pas de discréétisation

### Domaine 2 Problème 3 Maillage Quadrangle



On peut maintenant afficher la solution calculée en 3D pour visualiser la solution.

Solution pour le Domaine 2 Problème 3



Solution pour le Domaine 2 Problème 3

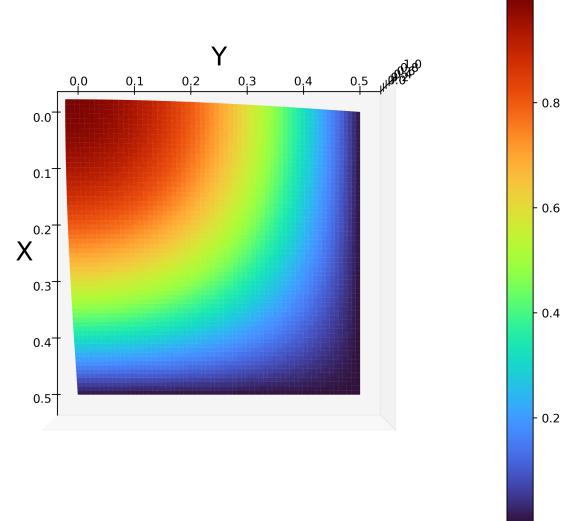


FIGURE 6 – Solution calculée vue sous différents angles Domaine 2 Problème 3

## 4 Quelques commentaires et analyses

On peut déjà constater que d'une manière générale, le code pour un maillage de type quadrangle est plus précis que pour un maillage de type triangle. En effet cela peut être expliqué par le fait que le schéma de quadrature est d'ordre 3 pour le quadrangle et d'ordre 2 pour le triangle (ordre 3 aussi pour le segment).

Globalement on observe un ordre 2 que ce soit pour un maillage de type triangle ou quadrangle. Dans certain cas, on est quand même assez en dessous de 2 (Problème 1 et 2 du Domaine 2 3.1 3.2).

Par ailleurs, on peut aussi remarquer que sur le domaine 2, on voit qu'un point remonte lorsque le maillage devient de plus en plus petit, on interpole donc que sur les points précédents et on ne prend pas encore compte ce dernier afin d'avoir une interpolation correct.

Cela peut être expliqué par le fait que le maillage devient tellement précis que l'erreur d'arrondi devient plus grande que l'erreur due à l'approximation du modèle. Afin d'affiner le calcul il conviendrait de passer à des nombres en double précision, on travaille ici en simple précision. La rigueur et la justesse des calculs pourrait être également revu.

On pourrait également utiliser des éléments de référence d'ordre supérieur comme les triangles ou quadrangle d'ordre 2.

Le code peut être encore amélioré dans sa rapidité sa justesse et optimisation mais également par la prise en compte de maillage 3D, à voir peut-être l'année prochaine.

**Fin de compte rendu.**

**Matthieu Petit / Alexandre Prampart**