Projet éléments finis

Lucien Gontier

December 18, 2023

1 Introduction

Ce projet vise à implémenter une méthode de résolution d'équations aux dérivées partielles par les éléments finis.
Nous nous intérésserons à la résolution du problème :

$$\begin{cases} -\Delta u & = f \quad \text{pour}(x,y) \in \Omega, \\ u(x,y) & = 0 \quad \text{pour } (x,y) \in \partial\Omega, \end{cases}$$

avec $\Omega = [0;1] \times [0;1]$, $\partial\Omega$ le bord de Ω, et $f \in L^2(\Omega)$ donnée.

2 Partie théorique

Nous allons tout d'abord faire l'étude théorique du problème avant de se lancer dans la programmation.

2.1 Formulation variationnelle et existence et unicité de u

Soit $\Omega = (0;1) \times (0;1)$, nous avons

$$\begin{cases} -\Delta u & = f \quad \text{pour}(x,y) \in \Omega, \\ u(x,y) & = 0 \quad \text{pour } (x,y) \in \partial\Omega, \end{cases}$$

• Soit V un espace de fonctions tests que l'on déterminera plus tard. Soit $v \in V$,

$$\int_\Omega -v\Delta u \, dx = \int_\Omega fv \, dx$$

Donc par les formules de Greens :

$$\int_\Omega \nabla u.\nabla v \, dx - \int_{\partial\Omega} (\nabla u.\tilde{n})v \, d\sigma = \int_\Omega fv \, dx$$

Si l'on choisit $V = H_0^1(\Omega)$, alors nous avons:

$$a(u,v) = l(v)$$

avec $a(u,v) := \int_\Omega \nabla u.\nabla v \, dx$ et $l(v) := \int_\Omega fv \, dx$

Nous aurons donc le problème variationnel suivant :
Trouver $u \in H_0^1(\Omega)$ tel que :

$$\forall v \in H_0^1(\Omega) \qquad a(u,v) = l(v)$$

Pour montrer l'existence et l'unicité d'une telle solution nous allons utiliser le théorème de Lax-Milgram.
• $H_0^1(\Omega)$ est un espace de Hilbert car c'est un sous espace fermé de $H^1(\Omega)$ qui est un espace de Hilbert.

• Montrons que l est une application linéaire continue.
l est linéaire par linéarité de l'intégrale.

De plus l est continue car :
$\forall v \in H_0^1(\Omega)$:

$$\begin{aligned} |l(v)| &= \left| \int_\Omega fv \, dx \right| \\ &\leq \|f\|_{L^2(\Omega)} \|v\|_{L^2(\Omega)} \text{ par cauchy Schwarz} \\ &\leq \|f\|_{L^2(\Omega)} \|v\|_{H^1(\Omega)} \text{ par définition de } \|.\|_{H^1(\Omega)} \end{aligned}$$

• Montrons que a est une application bilinéaire continue et coercive.
a est bilinéaire par linéarité de l'intégrale et du gradient.
a est continue car, $\forall u, v \in H_0^1(\Omega)$:

$$\begin{aligned} |a(u,v)| &= \left| \int_\Omega \nabla u.\nabla v \, dx \right| \\ &\leq \|\nabla u\|_{L^2(\Omega)} \|\nabla v\|_{L^2(\Omega)} \text{ par Cauchy-Schwarz} \\ &\leq \|u\|_{H^1(\Omega)} \|v\|_{H^1(\Omega)} \text{ par définition de } \|.\|_{H^1(\Omega)} \end{aligned}$$

Pour finir a est coercive car $\forall u \in H_0^1(\Omega)$:

$$\begin{aligned} a(u,u) &= \int_\Omega \nabla u.\nabla u \, dx \\ &= \|\nabla u\|^2_{L^2(\Omega)} \\ &\geq \frac{1}{2} \|\nabla u\|^2_{L^2(\Omega)} \frac{1}{2c} \|\nabla u\|^2_{L^2(\Omega)} \text{ par l'inégalité de Poincaré} \\ &\geq \frac{1}{2} max\left(1, \frac{1}{c}\right) \|u\|^2_{H^1(\Omega)} \text{ par définition de } \|.\|_{H^1(\Omega)} \end{aligned}$$

Donc par le théorème de Lax-Milgram, $\exists! u \in H_0^1(\Omega)$ tel que :

$$\forall v \in H_0^1(\Omega) \qquad a(u,v) = l(v)$$

Nous souhaitons maintenant retrouver cette solution sur Ω.
Nous supposons Ω 2 régulier, dans ce cas comme $u \in H_0^1(\Omega)$ solution du problème variationnel et $f \in L^2(\Omega)$, nous avons donc $u \in H^2(\Omega)$.
Soit $\varphi \in \mathcal{D}(\Omega) \subset H_0^1(\Omega)$, nousa vons donc :

$$\int_\Omega \nabla u.\nabla \varphi \, dx = \int_\Omega f\varphi \, dx$$

Par la formule de la dérivation faible nous obtenons :

$$\int_\Omega -\varphi \Delta u \, dx = \int_\Omega f\varphi \, dx$$

Donc :

$$\int_\Omega (-\Delta u - f)\varphi \, dx = 0$$

Donc $-\Delta u - f \in \mathcal{D}(\Omega)^{\perp_{L^2(\Omega)}}$
Or $\mathcal{D}(\Omega)^{\perp_{L^2(\Omega)}} = 0$, car $\overline{\mathcal{D}(\Omega)} = L^2(\Omega)$
Donc

$$-\Delta u = f, \text{ presque partout sur } \Omega$$

De plus on a $u \in H_0^1(\Omega)$, donc :

$$u = 0 \text{ presque partout sur } \partial\Omega$$

2.2 Rappel Méthode

2.2.1 Méthode de Galerkin

Nous avons montré qu'il esxiste une unique solution u presque partout sur Ω, nous souhaitons maintenant discrétiser le problème. Pour cela nous allons utiliser la méthode de Galerkin.
$V = H_0^1(\Omega)$ est un espace de Hilbert séparable car sous espace fermé de $H^1(\Omega)$, lui même Hilbert séparable.
Donc V admet une base hilbertienne, que nous noterons $(v_j)_{j\geq 1}$.
La méthode de Galerkin consiste à fixer un entier M, et à approcher V par $V_h = Vect\{v_j, 1 \leq j \leq M\}$, donc au lieau de considérer le problème variationnel, on considère le problème approché :
Trouver $u_h \in V_h$ tel que,

$$a(u_h, v_h) = l(v_h) \quad \forall v_h \in V_h$$

On note $u_{h,j}$ les composantes de u_h dans la base V_h, c'est à dire, $u_h = \sum_{j=1}^{M} u_{h,j} v_j$ Donc le problème se réecrit :
Trouver $(u_{h,j})_{1\leq j\leq M}$ tels que,

$$a\left(\sum_{j=1}^{M} u_{h,j} v_j, v_h\right) = l(v_h) \quad \forall v_h \in V_h$$

ou encore par linéarité de l, de a sur la seconde variable et la décomposition de tout v_h sur V_h :

$$a\left(\sum_{j=1}^{M} u_{h,j} v_j, v_k\right) = l(v_k) \quad \forall\, 1 \leq k \leq M$$

Par linéarité de a sur la première variable, nous retrouvons :

$$\sum_{j=1}^{M} u_{h,j} a(v_j, v_k) = l(v_k) \quad \forall\, 1 \leq k \leq M$$

Nous retrouvons donc un système linéaire à M équations pouvant s'écrire : Trouver le vecteur U tel que,

$$AU = L$$

avec $A := (a(v_j, v_k))_{1\leq j,k\leq M}$ matrice de taille $M \times M$, et L le vecteur de taille M des évalutations l aux différents éléments de la base de V_h.
La matrice A est inversible car l'application $a(.,.)$ est coercive.

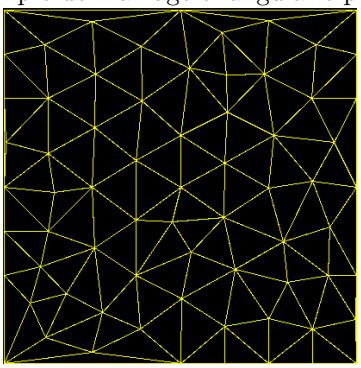
2.2.2 Méthode des éléments finis

La méthode des éléments finis s'inclut dans le cadre générale de la méthode de Galerkin, dans ce cas on se restreint à des approximations continues et polynomiales par morceaux, on choisit $(T_k)_k$ une famille de partie compacte connexe qui recouvre Ω (dans notre cas ce sera des triangles).
Pour chaque T_k, on lui associe $\mathbb{P}$ un espace de vectoriel de fonctions polynomiales de T_k dans $\mathbb{R}$, et $\Sigma = \sigma_i : \mathbb{P} \to \mathbb{R}, 1 \leq i \leq M$, une famille de M forme linéaire sur $\mathbb{P}$.
Mais tout d'abord définissons ce qu'est un élément fini.

On dit que le triplet $(T_k, \mathbb{P}, \Sigma)$ est un élément fini si Σ est $\mathbb{P}$ unisolvante.
Il est montré aussi que si $dim\mathbb{P} = card\Sigma$, et que $(T_k, \mathbb{P}, \Sigma)$ admet une base canonique alors $(T_k, \mathbb{P}, \Sigma)$ est un élément fini, c'est cette approche que nous utiliserons.
Nous nous placerons dans les éléments finis dits de Lagrange, c'est à dire $\forall 1 \leq i \leq M$ on a $\sigma_i(p) = p(s_i)$, avec s_i un point de T_k et $p \in \mathbb{P}$.
Dans notre cas nous décomposerons Ω en 2-simplexe de $[0,1] \times [0,1]$, (c'est à dire en triangle).

Nous prendrons $\mathbb{P}_1 = \{1, x, y\}$ ou $\mathbb{P}_2 = \{1, x, y, x^2, xy, y^2\}$ comme base de l'espace de fonction polynômiale.

Figure 1: exemple de maillage triangulaire pour [0,1]x[0.1]



2.3 Élements finis pour $\mathbb{P}_1$

On va faire l'étude de notre problème pour des éléments de type $\mathbb{P}_1$ c'est à dire avec des base de fonction polynomiale $\mathbb{P}_1$ dans nos éléments finis.
Pour cela on va se donner le triangle de référence $\hat{K}$, de sommets $\hat{s_1} = (0,0)$, $\hat{s_2} = (1,0)$ et $\hat{s_3} = (1,0)$.
Nous remarquons que $dimP = card\Sigma$, et si de plus nous prenons la famille de fonction polynomiale:

$$\begin{aligned}\hat{\varphi}_1(x,y) &= 1 - x - y\\ \hat{\varphi}_2(x,y) &= x\\ \hat{\varphi}_3(x,y) &= y\end{aligned}$$

définit une base canonique de $(\hat{K}, \mathbb{P}_{\hat{k}}, \Sigma_{\hat{K}})$
Donc $(\hat{K}, \mathbb{P}_{\hat{k}}, \Sigma_{\hat{K}})$ est un élément fini.
$\hat{K}$ sera notre triangle de référence, nous pourons passer à n'importe quel triangle du maillage par une transformation affine.
la composition par transformation affine conserve le fait que ce soit un élément fini, donc tous nos triangles sont des éléments finis.
Pour calculer cette transformation affine du triangle $\hat{K}$ vers un triangle K de sommet $s_1 = (x_1, y_1)$,$s_2 = (x_2, y_2)$ et $s_3 = (x_3, y_3)$, nous allons résoudre le système :

$$\begin{cases} B\hat{s_1} & +b = s_1\\ B\hat{s_2} & +b = s_2\\ B\hat{s_3} & +b = s_3\end{cases}$$

avec B une matrice 2x2 et b un vecteur de dimension 2.
Après résolution du système, nous obtenons :

$$B = \begin{pmatrix} x_2 - x_1 & x_3 - x_1\\ y_2 - y_1 & y_3 - y_1\end{pmatrix}, \qquad b = \begin{pmatrix} x_1\\ y_1\end{pmatrix}$$

Nous noterons $\mathcal{F}_i$ cette transposition associé au triangle K_i du maillage.
Elle est inversible car B l'est, on oeut donc écrire. Et nous aurons :

$$\begin{aligned}\varphi_1^i(x,y) &= \hat{\varphi}_1 \circ \mathcal{F}_i^{-1}(x,y)\\ \varphi_2^i(x,y) &= \hat{\varphi}_2 \circ \mathcal{F}_i^{-1}(x,y)\\ \varphi_3^i(x,y) &= \hat{\varphi}_3 \circ \mathcal{F}_i^{-1}(x,y)\end{aligned}$$

2.3.1 Résolution du système

Nous cherchons toujours à résoudre le système $AU = L$.
Soit M le nombre de noeuds dans le maillages, on $(\varphi_j)_{1\leq j\leq M}$ représente la base de V_h. • Nous allons

d'abord travailler sur L.
Nous avons que

$$L = \begin{pmatrix} l(\varphi_1) \\ \vdots \\ l(\varphi_M) \end{pmatrix}$$

Donc nous cherchons $\forall 1 \leq j \leq M$, quelle est la valeur de $l(\varphi_j)$.
On a :

$$\begin{aligned}
l(\varphi_j) &= \int_\Omega f\varphi_i \, dxdy \\
&= \sum_{K_i \in T_h} \int_{K_i} f\varphi_j \, dxdy \\
&= \sum_{K_i \in T_h, s_j \in K_i} \int_{K_i} f\varphi_j \, dxdy \\
&= \sum_{K_i \in T_h, s_j = s_1^i} \int_{K_i} f\varphi_1^i \, dxdy \quad + \sum_{K_i \in T_h, s_j = s_2^i} \int_{K_i} f\varphi_2^i \, dxdy \quad + \sum_{K_i \in T_h, s_j = s_3^i} \int_{K_i} f\varphi_3^i \, dxdy \\
&= \sum_{K_i \in T_h, s_j = s_1^i} \int_{\hat{K}} f \circ \mathcal{F}_i \, \hat{\varphi}_1 |det(J_{\mathcal{F}_i})| \, d\hat{x}d\hat{y} \\
&\quad + \sum_{K_i \in T_h, s_j = s_2^i} \int_{\hat{K}} f \circ \mathcal{F}_i \, \hat{\varphi}_2 |det(J_{\mathcal{F}_i})| \, d\hat{x}d\hat{y} \\
&\quad + \sum_{K_i \in T_h, s_j = s_3^i} \int_{\hat{K}} f \circ \mathcal{F}_i \, \hat{\varphi}_3 |det(J_{\mathcal{F}_i})| \, d\hat{x}d\hat{y}
\end{aligned}$$

Avec $J_{\mathcal{F}_i}$) la jacobienne asscoié à la transformation affine $\mathcal{F}_i$.
Donc par un changement de variable et du fait que $\varphi_1^i(x,y) = \hat{\varphi}_1 \circ \mathcal{F}_i^{-1}(x,y)$, on se retouve dans notre triangle de références.

• Nous allons faire la même chose pour la matrice A. Nous cherchons à connaître l'élement de ligne i et de colonne j de la matrice A, avec $1 \leq i, j \leq M$.

Nous voulons donc calculer, $a(\varphi_i, \varphi_j)$:

$$
\begin{aligned}
a(\varphi_i,\varphi_j) &= \int_\Omega \nabla\varphi_i.\varphi_j \; dxdy \\
&= \sum_{K_l\in T_h} \int_{K_l} \nabla\varphi_i.\varphi_j \; dxdy \\
&= \sum_{K_l\in T_h/s_i,s_j\in K_l} \int_{K_l} \nabla\varphi_i.\varphi_j \; dxdy \\
&= \sum_{K_l\in T_h/s_i=s_1^l,s_j=s_1^l} \int_{K_l} |\nabla\varphi_1^l|^2 dxdy + \sum_{K_l\in T_h/s_i=s_1^l,s_j=s_2^l} \int_{K_l} \nabla\varphi_1^l.\nabla\varphi_2^l dxdy + \sum_{K_l\in T_h/s_i=s_1^l,s_j=s_3^l} \int_{K_l} \nabla\varphi_1^l.\nabla\varphi_3^l dxdy \\
&\quad + \quad \ldots \quad + \quad \ldots \quad + \quad \ldots \qquad \text{pour } s_i = s_2^l \\
&\quad + \quad \ldots \quad + \quad \ldots \quad + \quad \ldots \qquad \text{pour } s_i = s_3^l \\
&= \sum_{K_l\in T_h/s_i=s_1^l,s_j=s_1^l} \int_{\hat{K}} |\left(J_{\mathcal{F}_i}{}^{-1}\right)^T \nabla\varphi_1^l|^2 |det(J_{\mathcal{F}_i})| d\hat{x}d\hat{y} \\
&\quad + \sum_{K_l\in T_h/s_i=s_1^l,s_j=s_2^l} \int_{\hat{K}} \left(\left(J_{\mathcal{F}_i}{}^{-1}\right)^T \nabla\varphi_1^l\right).\left(\left(J_{\mathcal{F}_i}{}^{-1}\right)^T \nabla\varphi_2^l\right) |det(J_{\mathcal{F}_i})| d\hat{x}d\hat{y} \\
&\quad + \sum_{K_l\in T_h/s_i=s_1^l,s_j=s_3^l} \int_{\hat{K}} \left(\left(J_{\mathcal{F}_i}{}^{-1}\right)^T \nabla\varphi_1^l\right).\left(\left(J_{\mathcal{F}_i}{}^{-1}\right)^T \nabla\varphi_3^l\right) |det(J_{\mathcal{F}_i})| d\hat{x}d\hat{y} \\
&\quad + \quad \ldots \qquad \text{pareil pour i=2,3}
\end{aligned}
$$

De la même manière que pour L, on arrive à écrire les éléments de A de manière à se retrouver dans $\hat{K}$. Maintenant que nous avons tout cela à notre disposition, nous pouvons attaquer le calcul approché de U.

3 Partie pratique

Nous avons étudié le problème de manière à pouvoir l'implémenter, pour cela nous aurons besoins de plusieurs choses

3.1 triangle

Pour décomposer notre espace en maillage nous utilisons le programme triangle mise au point par Jonathan Richard Shewchuk.
Ce programme nous permet de décomposer notre espace en différents type de maillage, ici nous utiliserons des triangles.
On peut choisir l'air maximal des triangle par l'option $-rpa$, et en fouillant la documentation lié au porogramme, j'ai vu que l'on pouvait utiliser l'option $-o2$ pour générer des éléments fini d'ordre 2.
Ce programme, nous renvoie deux 3 fichiers, nous nous intéresserons aux .ele et .node créés.
Dans le ".ele", nous retrouvons le nombre de triangle, le nombre de noeuds par triangle, et les numéros noeuds assoviés à chaque triangles.
Dans le ".node", nous retrouverons le nombre de noeuds, la dimensions de l'espace, nous y retrouverons pour chaque noeuds ses coordonnées spatiales et un entier 0 ou 1, pour savoir si il est oui ou non sur un des bords (1 sur le bord, 0 à l'intérieur).

3.2 Programme

3.3 recupération des éléments et des noeuds

J'ai donc crée un module comportant deux subroutines qui vont parcourir ces fichiers, récupérant le nombres d'éléments, le nombre de noeuds par élements, un tableau des éléments avec les noeuds associé , le nombre de noeuds, la dimension (pas nécessaire car on reste en dimension 2), un tableau

comportant les coordonées de chaque noeuds et un autre tableau récupérant 1 ou 0 selon si le noeuds est sur le bord.
Ensuite dans un autre module, j'ai créé deux subroutines pour initialiser $\hat{K}$, et pour créer un tableau qu sera de la taille (2*(nombre d'élements par noeuds), nombre d'éléments) qui permettra de stocker les éléments avec les coordonées des noeuds qui lui sont intérieurs.

3.3.1 Quadrature

Nous aurons besoins d'intégrer sur $\hat{K}$, c'est pourquoi j'ai créé un module contenant une subroutine faisant la quadrature associé.
L'idée pour $\mathbb{P}_1$ est d'évaluer la fonctions sur les trois sommets du triangles, que l'on multipliera par $\frac{1}{6}$, que l'on sommera ensuite pour obtenir une valeur approché de notre intégrale, cette quadrature nous permet une approximation exacte pour des fonctions polynomiales de degrés inférieurs ou égal à 1.

3.3.2 Fonctions

Nous aurons besoins de coder les $\hat{\varphi}$, leur gradient $\nabla\hat{\varphi}$, les transformation affines $\mathcal{F}_i$, les determinants de $J_{\mathcal{F}_i}$, les matrices $\left(J_{\mathcal{F}_i}{}^{-1}\right)^T$, $\forall 1 \leq i \leq nÉlements$, ainsi que la fonction f du problème, et une fonction U_{ex}, qui nous permettra de vérifier nos résultats. Tous ces éléments se retrouve dans le module "fonctions".
Pour U_{ex}, j'ai pris $U(x,y) = sin(2\pi x)sin(2\pi y)$, qui donne $f(x,y) = 8sin(2\pi x)sin(2\pi y)$, ou encore $U(x,y) = x(x-1)y(y-1)$, qui donne $f(x,y) = -2(x(x-1)+y(y-1))$.

3.3.3 Création de L et A

Maintenant que nous avons tous nos éléments, il nous faut créer L et A pour pouvoir résoudre notre système.
Nous allons tout d'abord, exclure les termes de bords car nous les ajouterons à la fin avec la conditions initiale.
Pour L, notre calcul de tout à l'heure permet en effet de donner L, cependant, si l'on veut savoir à quel noeud est raccorder chaque noeuds du maillages, il nous faudrait précalculer un tableau listant les noeuds et leur noeuds voisins, cela est couteux en terme de calcul.
Une idée plus simple est de parcourir tous les triangles, dans chaque triangle d'indice l, on va regarder ses noeuds, récupèrer leur indice global noté i associé à leur indice local mu au triangle, vérfier qu'ils ne soient pas sur le bords, si il ne le sont pas,on va ajouter la valeur de l'intégrale correspondantes à la valeur à l'indice i du tableau L.
Ensuite nous supprimerons les lignes du tableau dont l'indice i du noeuds associé se trouve sur le bord.
Nous nous retrouverosn avec un tableau de dimension dimL=(nombre de Noeuds - nombre de Noeuds sur le bord).

Nous procédons quasiment de la même manière pour A, sauf que nous ferons varier i, j à l'intérieur de chaque triangles pour obtenir toutes les combinaisons sur i,j=1,2,3 en noeuds locales, si l'un des deux noeuds n'est pas sur le bords alors, nous ajouterons la valeur de l'intégrale à la ligne i, colonne j de la matrice A.
Ensuite on vient retirer les lignes dont les noeuds sont sur le bord, pusi on fait la même chose pour les colonnes.
Nous retrouvons bien une matrice de taille (dimL x dimL).

3.3.4 Résolution du problème

Pour résoudre le problème, j'ai ensuite crée un module appelant la subroutine DGESV de la librairie LAPACK, permettant de résoudre le système en passant par une décomposition LU. Ensuite je viens intercaler des 0 aux indices des noeuds du bords, et je retrouve l'approximation de U sur tous les noeuds de mon maillage.

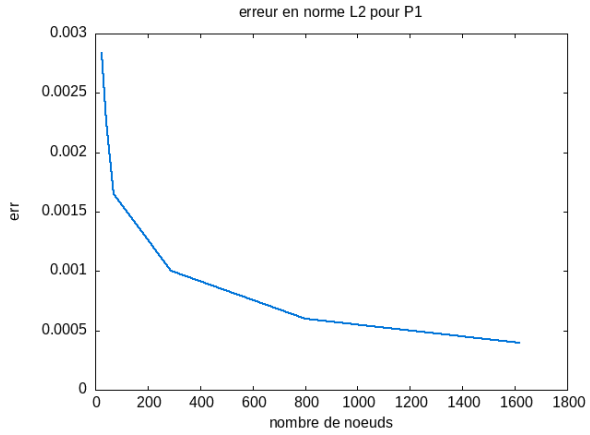
3.3.5 Tracé de U

Une fois U récupérer on construit un fichier .vtk qui nous permettra de le tracer sur paraview, de même pour un fichier avec la valeur exacte de U sur les noeuds, qui nous permettra de comparer visuellement les deux solutions.

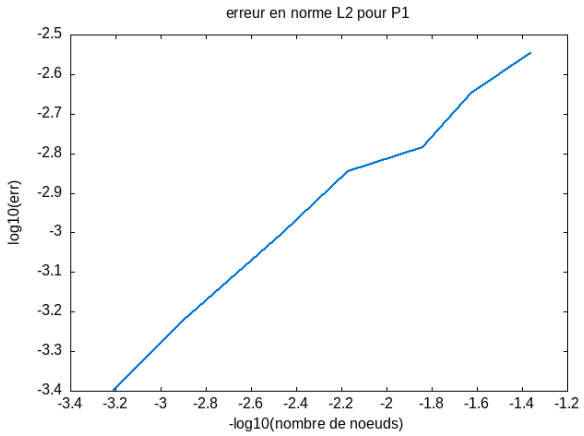
3.3.6 Erreur

Une autre subroutine permet de calculer l'erreur en norme L^2 de la solution approché, je reécupère ensuite les données que je mets dans un fichier, qui me permet de tracer l'erreur en fonction du nombre de noeuds, et de regarder l'ordre de notre approximation.

Nous obtenons une belle convergence des erreurs:



et maintenant en échelle logarithmique pour récupérer l'ordre :



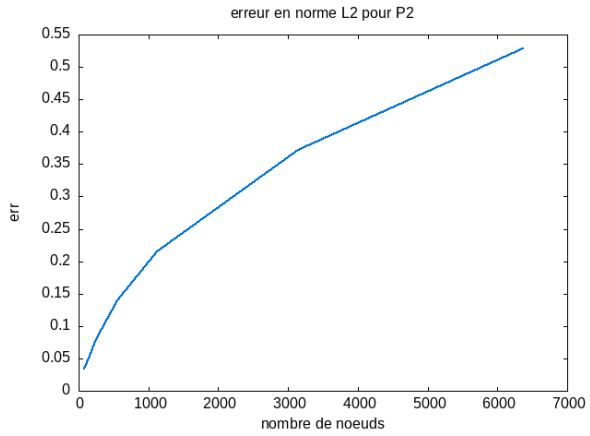
Cela nous donne un ordre de envrion 1,25.

4 Méthode P2

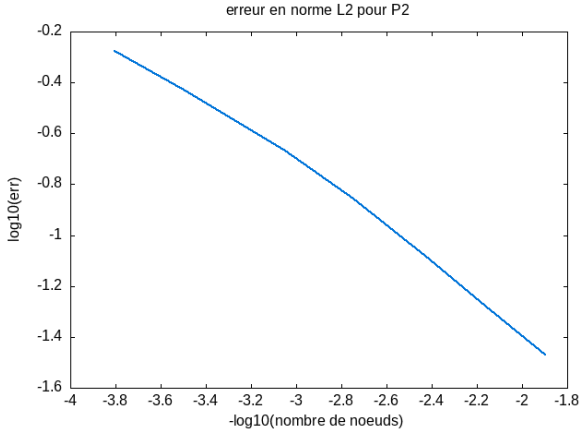
Tout mon code est adaptable en pour $\mathbb{P}_2$ il suffit de choisir 2 en première ligne du fichier "data/init.dat", et de choisir un nom de fichier avec P2 dedans dans ceux disponible dans le dossier "triangle". Pour cela j'ai adapté tout mon code avec des select case lorsqu'il y en avait besoin, par exemple pour l'initialisation de $\hat{K}$ ou les fonctions pour le calcul de φ et de $\nabla\varphi$, que j'ai spécifiquement recalculé dans le cas $\mathbb{P}_2$.

Cependant l'erreur obtenue diverge en augmentant le nombre de noeuds, peut être cela vient de ma quadrature, qui reste celle où je calcule la valeur de la fonctions sur tous les noeuds, et je divise ensuite par 2*(nombre de noeuds). Je voulais en implémenter une autre qui utilise 6 noeuds intérieur au

triangle, mais je n'ai malheuresement pas eu le temps de le faire.
Voici les graphes d'erreurs pour $\mathbb{P}_2$.



et maintenant en échelle logarithmique pour récupérer l'ordre :



On trouve un ordre négatif, ce qui n'est pas du tout bon signe.
De plus je ne peux pas vérifier mes résultats car paraview n'arrive pas à lire mon fichier vtk avec 6 noeuds par éléments.