Application de la méthode des Eléments Finis

Cas $\mathbb{P}_1$ Lagrange 1D

UFR des Sciences Montpellier

Master 1 Modélisation et Analyse Numérique (MANU)

Djebra Rabah
Messaoudi Feriel

Date : 6 octobre 2024

Table des matières

Table des figures

1 Introduction

Ce projet porte sur la résolution d'un problème aux limites dans le cadre des équations différentielles. Nous examinons un problème défini sur un domaine Ω, où il s'agit de trouver une fonction u qui satisfait une équation différentielle spécifique ainsi que des conditions aux limites imposées.

L'objectif principal de ce projet est de démontrer l'existence et l'unicité de la solution pour ce problème. Dans la première partie, nous nous concentrons sur la preuve théorique de l'existence et de l'unicité de la solution.

Dans la deuxième partie, nous nous intéressons à la discrétisation du problème en utilisant la méthode des éléments finis. Nous appliquons cette méthode avec une discrétisation uniforme de l'intervalle Ω, pour les cas 1D.

2 Position de Problème

On considère le problème aux limites suivant :

$$\begin{cases} -\Delta u + au = f & dans\ \Omega \\ \nabla u \cdot n = \alpha & dans\ \partial\Omega \end{cases} \tag{1}$$

où $f \in L^2(\Omega),\ a \in L^\infty(\Omega),\ \alpha \in L^2(\partial\Omega)$.

3 Existance et unicité de solution

3.1 Formulation variationnelle

On considère un espace fonctionnelle quelconque X, on prend une fonction test $v \in X$ et on multiplier par v l'EDP associée à (1) et on integre sur Ω.

$$\int_\Omega (-\Delta u + au)vdx = \int_\Omega fvdx$$

par une integration par partie :

$$-\int_{\partial\Omega} (\nabla u \cdot n)vds + \int_\Omega \nabla u \cdot \nabla vdx + \int_\Omega auvdx = \int_\Omega fvdx$$

Et d'aprés la condition de neumann de (1) on trouve :

$$\int_\Omega \nabla u \cdot \nabla vdx + \int_\Omega auvdx = \int_\Omega fvdx + \int_{\partial\Omega} \alpha vds.$$

Comme $f \in L^2(\Omega)$ alors $v \in L^2(\Omega)$ donc sur le coté droit de la formule il faut que $\nabla v \in L^2(\Omega)$ d'où $v \in H^1(\Omega) = X$.

le problème (1) se récrit sous la forme variationnelle suivante :

$$\begin{cases} \text{Trouver } u \in H^1(\Omega) \text{ tel que :} \\ \displaystyle\int_\Omega \nabla u \cdot \nabla v dx + \int_\Omega auvdx = \int_\Omega fvdx + \int_{\partial\Omega} \alpha v ds \quad \forall\, v \in H^1(\Omega). \end{cases} \tag{2}$$

On note $a(u,v) = \int_\Omega \nabla u \cdot \nabla v dx + \int_\Omega auvdx$
Et $l(v) = \int_\Omega fvdx + \int_{\partial\Omega} \alpha v ds$

3.2 Vérification les hypothèse de Lax-Milgram

Pour montrer que (2) admet une unique solution, il faut verifie les condition de (Théorème de Lax-Milgram) :

Théoreme :

Soit H un espace de Hilbert, soit $a(\cdot,\cdot) : H \times H \longrightarrow \mathbb{R}$ une forme bilinéaire, continue, et coercive

$$\exists \alpha\ ,\ C > 0, \text{tel que } |a(u,v)| \leq C\|u\|\|v\| \text{ et } a(u,u) \geq \alpha\|u\|^2,\ \forall u, v \in H,$$

Et soit l une forme linéaire et continue sur H.

Alors il existe un unique $u \in H$ tel que $\forall v \in H, a(u,v) = l(v)$.

De plus, si a est symétrique, alors u est l'unique minimum de

$$J : v \in H \mapsto \frac{1}{2}a(v,v) - l(v).$$

Remarque : Dans notre problème on remarque que a est symetrique.

- On a bien $H^1(\Omega)$ est un espace de hilbert.
- **Bilinéarité de a :**
  Comme a est symetrique il suffit de verifie la linearité à gauche de a, soit $u, v, w \in H^1(\Omega)$ et $\lambda \in \mathbb{R}$,

$$\begin{aligned} a(u+\lambda v, w) &= \int_\Omega \nabla(u+\lambda v) \cdot \nabla w dx + \int_\Omega a(u+\lambda v)w \\ &= \int_\Omega \nabla u \cdot \nabla w dx + \lambda \int_\Omega \nabla v \cdot \nabla w dx \\ &= a(u,w) + \lambda a(v,w). \end{aligned}$$

  Puisque a est symetrique, alors a est lineaire à droite, donc a est bilinéaire.

- **La continuité de a :**
  Par l'inégalité triangulaire :

$$|a(u,v)| = \left|\int_\Omega \nabla u \cdot \nabla v dx + \int_\Omega auvdx\right| \leq \left|\int_\Omega \nabla u \cdot \nabla v dx\right| + \left|\int_\Omega auvdx\right|$$

  On utilise l'inégalité de cauchy schwarz et l'hypothèse de $a(x) \in L^\infty(\Omega)$:

$$|a(u,v)| \leq \|\nabla u\|_{L^2(\Omega)}\|\nabla v\|_{L^2(\Omega)} + \|a\|_{L^\infty(\Omega)}\|u\|_{L^2(\Omega)}\|v\|_{L^2(\Omega)}$$

  Et par définition de la norme $H^1(\Omega), \|u\|_{H^1(\Omega)} = \|u\|_{L^2(\Omega)} + \|\nabla u\|_{L^2(\Omega)}$, d'où

$$|a(u,v)| \leq C\|u\|_{H^1(\Omega)}\|v\|_{H^1(\Omega)}$$

  avec $C > 0 = 1 + \|a\|_{L^\infty(\Omega)}$.
- **La coercivité de a :**
  On prend comme hypothèse $\min_{x\in(\Omega)} a(x) = a_0 > 0$ et par définition de la norme $L^2(\Omega)$ et norme $H^1(\Omega)$:

$$a(u,u) = \int_\Omega |\nabla u|^2 dx + \int_\Omega au^2 dx \geq \|\nabla u\|^2_{L^2(\Omega)} + a_0\|u\|^2_{L^2(\Omega)} \geq \beta\|u\|^2_{H^1(\Omega)}$$

  avec $\beta > 0 = \min_{x\in\Omega}(1, a_0)$.
- **La linearité de l :**
  Soit $u, v \in H^1(\Omega)$ et $\lambda \in \mathbb{R}$ avec l'hypothèse de $\alpha \in L^2(\partial\Omega)$, par linearité de l'integral :

$$\begin{aligned} l(u+\lambda v) &= \int_\Omega f(u+\lambda v)dx + \int_{\partial\Omega} \alpha(u+\lambda v)ds = \int_\Omega fudx + \lambda\int_\Omega fvdx + \int_{\partial\Omega}\alpha uds + \lambda\int_{\partial\Omega}\alpha vds \\ &= l(u) + \lambda l(v) \end{aligned}$$

  $\Longrightarrow l$ est lineaire.
- **La continuité de l :**
  On prend l'hypothèse $\alpha \in L^2(\partial\Omega)$ et par l'inégalité triangulaire :

$$|l(v)| = \left|\int_\Omega fvdx + \int_{\partial\Omega}\alpha vds\right| \leq \left|\int_\Omega fvdx\right| + \left|\int_{\partial\Omega}\alpha vds\right|$$

  Et par l'inégalité de cauchy schwarz et l'inégalité de trace qui assure $\exists\ C_t > 0$ tel que :

$$\|\alpha\|_{\partial\Omega} \leq C_t\|\alpha\|_{H^1(\Omega)}$$

  et par définition de norme $H^1(\Omega)$

$$|l(v)| \leq \|f\|_{L^2(\Omega)}\|v\|_{L^2(\Omega)} + \|\alpha\|_{L^2(\partial\Omega)}\|v\|_{L^2(\partial\Omega)} \leq C_\Omega\|v\|_{H^1(\Omega)}$$

  avec $C_\Omega > 0 = \|f\|_{L^2(\Omega)} + C_t\|\alpha\|_{L^2(\Omega)}$. Donc l est continue.

Les hypothèses du théorème de Lax-Milgram sont satisfaites, on peut conclure qu'il existe une unique solution $w \in H^1_0(0,1)$ pour le problème variatoinnelle (2).

3.3 L'équivalence avec le problème limite

Soit $u \in H^1(\Omega)$ solution de (2), au début on considère $u \in C^2(\Omega) \cap H^1(\Omega)$ avec $v \in D(\Omega)$,

$$\int_\Omega \nabla u \cdot \nabla v dx + \int_\Omega auv dx - \int_\Omega fv dx - \int_{\partial\Omega} \alpha v ds = 0 \tag{3}$$

Par une intégration par partie (3) devient :

$$\int_{\partial\Omega} (\nabla u \cdot n) v ds - \int_\Omega \Delta u v dx + \int_\Omega auv dx - \int_\Omega fv dx - \int_\Omega \alpha v ds \tag{4}$$

Comme $v \in \mathcal{D}(\Omega)$ alors $v_{|\partial\Omega} = 0$ donc (4) devient :

$$\int_\Omega (-\Delta u + au - f) v dx = 0$$

Vrai pour tout $v \in \mathcal{D}(\Omega)$ et comme $f \in L^2(\Omega)$ on aura :

$$-\Delta u + au = f \text{ presque partout dans } \Omega$$

On revient à (3) avec $v \in H^1(\Omega)$, pour avoir $-\Delta u + au = f$ sur Ω il faut :

$$\int_{\partial\Omega} ((\nabla u \cdot n) - \alpha) v ds = 0$$

Alors $\nabla u \cdot n = \alpha$ sur $\partial\Omega$, donc $u \in H^1(\Omega)$ est l'unique solution de problème limite (1)

4 Approximation Numérique avec FreeFem++ :

```
// Script FreeFem++ pour une approximation 1D :
int n = 10;
real L = 1.0;
real H = 1e-6;
mesh Th = square(n, 1, [L*x, H*y]);
fespace Vh(Th, P1);
Vh u, v;
real a = 1.0;
func f = x*(1-x);
problem Poisson1D(u, v) =
    int2d(Th)(dx(u)*dx(v) + dy(u)*dy(v) + a*u*v) - int2d(Th)(f*v)
    + on(0, u=0) + on(1, u=0);
Poisson1D;
plot(u, wait=true);
```

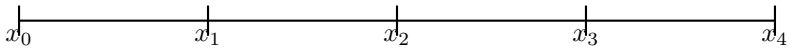
Listing 1 – Exemple de Script FreeFem++ pour une approximation 1D

```
// Script FreeFem++ pour une approximation 2D
int n = 10;
real L = 1.0;
mesh Th = square(n, n, [L*x, L*y]);
fespace Vh(Th, P1);
Vh u, v;
real a = 1.0;
func f = x*(1-x)*y*(1-y);
func alpha = 0.0;
problem Poisson2D(u, v) =
    int2d(Th)(dx(u)*dx(v) + dy(u)*dy(v) + a*u*v) - int2d(Th)(f*v)
    + on(0, u=0);
Poisson2D;
plot(u, wait=true);
```

Listing 2 – Exemple de Script FreeFem++ pour une approximation 2D

5 Élément finis $\mathbb{P}_1$-Lagrange

Dans cette partie, nous nous intéressons à l'approximation par élément finis dans le cas 1D Nous introduisons une discrétisassion de l'intervalle $\Omega = [0,1]$ en $N+1$ sous-intervalles. Nous fixons un entier $N > 1$, posons $h = \frac{1}{N+1}$



Intervalle Ω divisé en $N+1=4$ segments égaux

0n définit l'espace $V_h \subset H^1(\Omega)$ tel que :

$$V_h = \left\{ v_h \in C^0(\Omega), v_{h[x_i,x_{i+1}]} \in \mathbb{P}_1 \quad \forall\, i = 0, \ldots, N \right\}$$

Dans notre cas $\dim V_h = N+2$.

Rappellons que V_h est engendré par une base $\{\phi_i\}_{i=0,\ldots,N+1}$ tel que pour $i=0$ et $i=N+1$:

$$\phi_0(x) = \begin{cases} \dfrac{x_1 - x}{h} & \text{si } x \in [x_0, x_1] \\ 0 & \text{sinon} \end{cases}; \quad \phi_{N+1}(x) = \begin{cases} \dfrac{x - x_N}{h} & \text{si } x \in [x_N, x_{N+1}] \\ 0 & \text{sinon} \end{cases}$$

Et pour $i = 1, \ldots, N$:

$$\phi_i(x) = \begin{cases} \dfrac{x - x_{i-1}}{h} & \text{si } x \in [x_{i-1}, x_i] \\ \dfrac{x_{i+1} - x}{h} & \text{si } x \in [x_i, x_{i+1}] \\ 0 & \text{sinon} \end{cases}$$

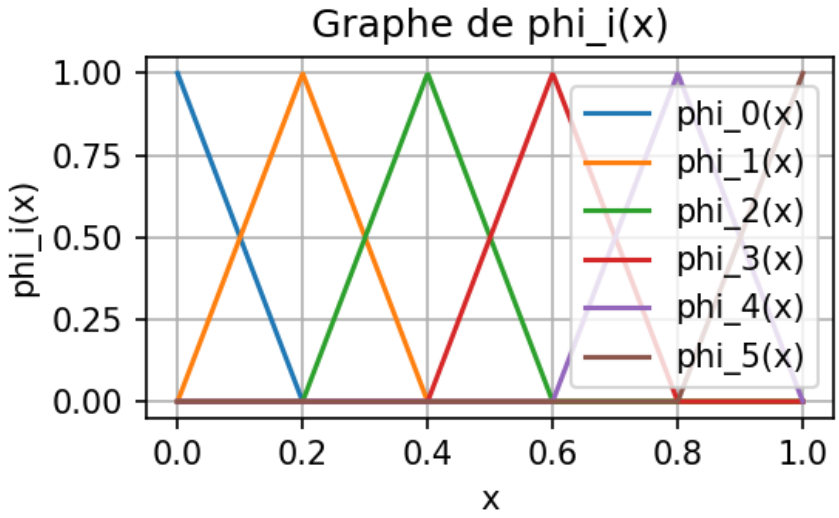


FIGURE 1 – Les fonctions chapeaux

De sorte que le problème variationnelle approché s'écrit :

$$\begin{cases} \text{Trouver } u_h \in V_h \text{ tel que :} \\ a(u_h, v_h) = l(v_h) \quad \forall\, v_h \in V_h. \end{cases} \tag{5}$$

On décompose u_h sur la base $\{\phi_i\}_{i=0,\dots,N+1}$, tel que :

$$u_h(x) = \sum_{i=0}^{N+1} u_h(x_i)\phi_i(x)$$

Comme $\{\phi_j\}_{j=0,\dots,N+1}$ une base de V_h , on impose $v_h(x) = \phi_j(x)$ donc :

$$a(u_h, v_h) = l(v_h) \Leftrightarrow a(\sum_{i=0}^{N+1} u_h(x_i)\phi_i, \phi_j) = l(\phi_j) \Leftrightarrow \sum_{i=0}^{N+1} u_h(x_i)a(\phi_i, \phi_j) = l(\phi_j)$$

Par bilinearité de a, équivaut à un systeme linéaire :

$$AU_h = C \Longleftrightarrow A_1U_h + B_1U_h = C$$

Les éléments de A_1, B_1, C sont donnés respectivement comme suit :

Avec $K_j = [x_i, x_{i+1}]$ pour $i \in 0, \ldots, N$:

$$a_{ij} = \int_{K_j} \nabla\phi_i(x) \cdot \nabla\phi_j(x)dx$$
$$b_{ij} = \int_{K_j} a(x_i)\phi_i(x)\phi_j(x)dx$$
$$C = \begin{pmatrix} -\alpha + \int_{x_0}^{x_1} f(x_0)\phi_0(x)dx \\ \int_{x_0}^{x_2} f(x_1)\phi_1(x)dx \\ \vdots \\ \int_{x_{N-1}}^{x_{N+1}} f(x_N)\phi_N(x)dx \\ +\alpha + \int_{x_N}^{x_{N+1}} f(x_{N+1})\phi_{N+1}(x)dx \end{pmatrix}_{(N+2)}$$

Il n'existe que deux fonctions de base ϕ_j et ϕ_{j+1} sur n'importe quelle maille K_j de Ω.

5.1 Propriété de la matrice A :

Soit $x \in \mathbb{R}^N$ non nul $x = (x_1, \ldots, x_N)^T$ avec les x_i tous non nuls et on prend $A = A_1 + B_1$, On aura :

— **Pour** A_1 **:**

$$x^T A_1 x = \sum_{i=0}^{N+1}\sum_{j=0}^{N+1} x_i x_j \int_\Omega \nabla\phi_i \nabla\phi_j dx = \int_\Omega \left(\sum_{i=0}^{N+1} x_i \nabla\phi_i\right)\left(\sum_{j=0}^{N+1} x_j \nabla\phi_j\right) dx$$

Posons $X_h = \sum_{i=0}^{N+1} x_i \nabla\phi_i$, on a alors $x^T A_1 x = \int_\Omega |X_h'|^2 dX = \|X_h'\|^2_{L^2(\Omega)} \geq 0$, car $\nabla\phi_i$ peuvent éventuellement être nulles pour certains $i \in [0, N+1]$. Donc A_1 est **semi-définie positive**.

— **Pour** B_1 **:**

$$x^T B_1 x = \sum_{i=0}^{N+1}\sum_{j=0}^{N+1} x_i x_j \int_\Omega \phi_i \phi_j dx = \int_\Omega \left(\sum_{i=0}^{N+1} x_i \nabla\phi_i\right)\left(\sum_{j=0}^{N+1} x_j \nabla\phi_j\right) dx$$

Posons $Y_h = \sum_{i=0}^{N+1} x_i \phi_i$, on a alors : $x^T B_1 x = \int_\Omega |X_h'|^2 dX = \|X_h'\|^2_{L^2(\Omega)} > 0$, car $x_i \neq 0$, $\forall i = 0, \cdots, N+1$ et les ϕ_i sont des bases de V_h ne sont pas uniformement nulles, donc $x^T B_1 x > 0$, d'ou B_1 est **définie positive**.

Alors A est **définie positive**.

5.2 Construction des matrices locales et globales

On note A_h, B_h les matrices locales associées à une maille fixée K_j , sachant que $mes(K_j) = h$.

Le terme général de A_h et B_h :

$$A_{h_{11}} = \int_{K_j} (\nabla\phi_i)^2 dx = \int_{K_j} (\frac{-1}{h})^2 dx = \frac{1}{h}$$
$$A_{h_{12}} = \int_{K_j} \nabla\phi_i \cdot \nabla\phi_{i+1} = \int_{K_j} (\frac{-1}{h})(\frac{1}{h}) dx = \frac{-1}{h} = A_{h_{21}} \text{ car } a \text{ est symetrique}$$
$$A_{h_{22}} = \int_{K_j} (\nabla\phi_{i+1})^2 dx = \int_{K_j} (\frac{1}{h})^2 dx = \frac{1}{h}$$

On prend $a(x) = a_0 > 0$.

$$B_{h_{11}} = \int_{K_j} a_0(\phi_i)^2 dx = \frac{a_0 h}{3}$$
$$B_{h_{12}} = \int_{K_j} a_0(\phi_i)(\phi_{i+1}) dx = \frac{a_0 h}{6} = B_{h_{21}} \text{ car } a \text{ symetrique}$$
$$B_{h_{22}} = \int_{K_j} a_0(\phi_{i+1})^2 dx = \frac{a_0 h}{3}$$

D'où

$$A_h = \tfrac{1}{h} \begin{pmatrix} 1 & -1 \\ -1 & 1 \end{pmatrix}; \quad B_h = \tfrac{a_0 h}{6} \begin{pmatrix} 2 & 1 \\ 1 & 2 \end{pmatrix}$$

On utilise les matrices locales A_h *et* B_h . On construit les matrices globales A_1, B_1 de taille $(N+2 \times N+2)$, données par :

$$A_1 = \tfrac{1}{h} \begin{bmatrix} 1 & -1 & 0 & \cdots & \cdots & 0 \\ -1 & 2 & -1 & \ddots & \ddots & \vdots \\ 0 & \ddots & \ddots & \ddots & \ddots & \vdots \\ \vdots & \ddots & \ddots & \ddots & \ddots & 0 \\ \vdots & \ddots & 0 & -1 & 2 & -1 \\ 0 & \cdots & \cdots & 0 & -1 & 1 \end{bmatrix}; \quad B_1 = \tfrac{a_0 h}{6} \begin{bmatrix} 2 & 1 & 0 & \cdots & \cdots & 0 \\ 1 & 4 & 1 & \ddots & \ddots & \vdots \\ 0 & \ddots & \ddots & \ddots & \ddots & \vdots \\ \vdots & \ddots & \ddots & \ddots & \ddots & 0 \\ \vdots & \ddots & 0 & 1 & 4 & 1 \\ 0 & \cdots & \cdots & 0 & 1 & 2 \end{bmatrix}$$

5.3 Estimation d'erreur et convergence

Pour analyser la convergence de la méthode des éléments finis, on utilise **Lemme de céa** qui donne une relation entre la solution exacte de problème variationnel et la solution approchée obtenu de problème variationnel approché.

Lemme de Céa : Sous les hypothèses de Lax-Milgram avec u la solution exacte de problème variationnel et u_h la solution de problème variationnel approché on a :

$$\|u - u_h\|_V \leq \frac{M}{\alpha} \inf_{v_h \in V_h} \|u - v_h\|_V$$

Où M : la constante de continuité de la forme bilinéaire $a(\cdot,\cdot)$ associée au problème variationnelle, et α : la constante de coércivité associée à $a(\cdot,\cdot)$. L'ordre de convergence en norme H^1 pour la méthode des éléments finis dépend de la régularité de la solution exacte et du degré des polynômes utilisés dans les éléments finis.

5.4 Exemple 1D avec python

```
import numpy as np
import matplotlib.pyplot as plt
from scipy.integrate import quad
from scipy import integrate


def phi_i(x, i, x_bar):
    h = x_bar[1] - x_bar[0]
    N = len(x_bar) - 1  # ici N = Nombre total d'intervalles
    if i == 0:
        if x_bar[0] <= x <= x_bar[1] :
            return (x_bar[1] - x) / h
        else :
            return 0
    elif i == N:
        if x_bar[-2] <= x <= x_bar[-1] :
            return (x - x_bar[-2]) / h
        else :
            return 0
    elif 1 <= i < N:
        if x_bar[i-1] <= x <= x_bar[i]:
            return (x - x_bar[i-1]) / h
        elif x_bar[i] <= x <= x_bar[i+1]:
            return (x_bar[i+1] - x) / h
        else:
            return 0
    else:
        return 0
def dphi_i(x, i, x_bar):
    h = x_bar[1] - x_bar[0]
    N = len(x_bar) - 1  # ici N = Nombre total d'intervalles
    if i == 0:
        if x_bar[0] <= x <= x_bar[1] :
            return -1/h
        else :
            return 0
    elif i == N:
        if x_bar[-2] <= x <= x_bar[-1] :
            return 1/h
        else :
            return 0
    elif 1 <= i < N:
        if x_bar[i-1] <= x <= x_bar[i]:
            return 1/h
        elif x_bar[i] <= x <= x_bar[i+1]:
            return -1/h
        else:
            return 0
```

```
    else:
        return 0

def second_membre(alpha, f, x_bar,N):
    C = np.zeros((N + 2, 1))

    for i in range(N + 2):
        if i == 0:
            p1 = lambda x: f(x_bar[0]) * phi_i(x, 0, x_bar)
            integral,_ = quad(p1, x_bar[0], x_bar[1])
            C[i] = -alpha + integral
        elif i == N + 1:
            p1 = lambda x: f(x_bar[N + 1]) * phi_i(x, N + 1, x_bar)
            integral,_ = quad(p1, x_bar[N], x_bar[N + 1])
            C[i] = +alpha + integral
        else:
            p1 = lambda x: f(x_bar[i]) * phi_i(x, i, x_bar)
            integral,_ = quad(p1, x_bar[i-1], x_bar[i+1])
            C[i] = integral

    return C
def Matrice_Local(a0, h):
    A_h = np.array([[(1/h) + (a0*h)/3, (-1/h) + (a0 * h) / 6], [(-1/h) + (a0 * h)/6,
        (1/h) + (a0*h)/3]])
    return A_h

def Matrice_Global(N, a0, h):
    A = np.zeros((N + 2, N + 2))
    A_h = Matrice_Local(a0, h)

    for i in range(N + 1):
        A[i:i+2, i:i+2] += A_h
    return A

def assemblage(N, a0, alpha, f):
    h = 1 / (N + 1)
    x_bar = np.linspace(0, 1, N + 2)
    C = second_membre(alpha, f, x_bar,N)
    A_h = Matrice_Local(a0, h)
    A = Matrice_Global(N, a0, h)
    return A_h,A, C
def gauss_seidel(A, b, tol=1e-10, max_iterations=1000):
    n = len(b)
    x = np.zeros(n)
    D = np.diag(np.diag(A))
    E = -np.tril(A, -1)
    F = -np.triu(A, 1)
    for k in range(max_iterations):
        x_avant = x.copy()
        x = np.dot(np.linalg.inv(D - E), np.dot(F, x) + b)
        err = np.linalg.norm(x - x_avant)
        if err < tol:
            print("La convergence atteinte  apr s",k+1,"it rations.")
            return x[:,0]
    print("Nombre maximal d'it rations atteint sans convergence.")
    return x
 # la soltion exacte de probl me :
```

```
def u(x):
    return (-3/2)*(x**3) + x**2 + x
    #return x
def u_prime(x):
    return -4.5 * x**2 + 2 * x + 1
    #return 1
N=4
a0 = 1.0
alpha = 1.0
x = np.linspace(0, 1, 500)    # pour tracer les phi_i
x_bar = np.linspace(0, 1, N + 2)
for i in range(N + 2):
    y = [phi_i(xi, i, x_bar) for xi in x]
    plt.plot(x, y, label=f'phi_{i}(x)')

plt.xlabel('x')
plt.ylabel('phi_i(x)')
plt.title('Graphe de phi_i(x)')
plt.legend()
plt.grid(True)
plt.show()
# Exemple
def Exemple(N) :
    h = 1/(N+1)
    a0 = 1.0
    alpha = 1.0
    x_bar = np.linspace(0, 1, N + 2)
    def f(x):
        #return x
        return (-2/3)*(x**3) + x**2 + 5*x - 2

    A_h,A, C = assemblage(N, a0, alpha, f)

    print("La matrice local A_h est :", A_h)
    print("La matrice global A est :", A)
    print("Le second_membre C:", C)
    u_h = gauss_seidel(A, C)
    print("Solution approch  du syst me lin aire:", u_h)
    u_exact = np.zeros(N+2)
    u_exact_prime = np.zeros(N+2)
    for i in range(N+2):
        u_exact[i]= u(x_bar[i])
        u_exact_prime[i] = u_prime(x_bar[i])
    print("la solution exact de probl me est :",u_exact)
    plt.plot(x_bar,u_h,'b+',x_bar,u_exact,'r')
    plt.legend(['sol_approche', 'sol_exact'])
    plt .title('graphe de comparaison')
    plt.show()
    e_L2 = np.zeros(N + 2)
    e_H1 = np.zeros(N + 2)
    u_h = u_h.flatten()
    for i in range(N + 1):  # Modifiez ici pour N + 1 au lieu de N + 2
        g_1 = lambda x: (u(x) - u_h[i]) ** 2
        g_2 = lambda x: (u_prime(x) - (u_h[i + 1] - u_h[i]) / h) ** 2
        e_L2[i] = integrate.quad(g_1, x_bar[i], x_bar[i + 1])[0]
        e_H1[i] = integrate.quad(g_1, x_bar[i], x_bar[i + 1])[0] + integrate.quad(g_2,
             x_bar[i], x_bar[i + 1])[0]
```

```
    err_L2 = np.sqrt(np.sum(e_L2))
    err_H1 = np.sqrt(np.sum(e_H1))
    print("Erreur L2 :", err_L2)
    print("Erreur H1 :", err_H1)

    return err_L2,err_H1
Exemple(5)
vect_n = [100, 150,200,250]
vect_err_l2 = []
vect_err_H1 = []
for N in vect_n:
    err_l2,err_H1 = Exemple(N)
    vect_err_l2.append(err_l2)
    vect_err_H1.append(err_H1)

h_vect = [(1)/(N+1) for N in vect_n]
print("le vecteur h =",h_vect)
print("le vecteur erreur L2 =",vect_err_l2)
print("le vecteur erreur H1 =",vect_err_H1)
plt.loglog(h_vect, vect_err_l2, 'bo-', h_vect,vect_err_H1,'ko-',h_vect,h_vect,'ro-',
    h_vect,[h_vect**2 for h_vect in h_vect],'go-')
plt.xlabel('h')
plt.ylabel('erreur')
plt.title('graphe de convergence')
plt.legend(['err_l2','err_H1','h','h^2'])
plt.show()
ordre1 = np.zeros(len(vect_n)-1)
ordre2 = np.zeros(len(vect_n)-1)
for i in range(len(vect_n)):
    ordre1[i-1]= (np.log(vect_err_l2[i-1]/vect_err_l2[i]))/(np.log(h_vect[i-1]/h_vect[
        i]))
    ordre2[i-1]= (np.log(vect_err_H1[i-1]/vect_err_H1[i]))/(np.log(h_vect[i-1]/h_vect[
        i]))
print("l'ordre de convergence en norme L2 est :",ordre1)
print("l'ordre de convergence en norme H1 est :",ordre2)
```

Listing 3 – Exemple de code Python cas 1D

6 Conclusion

Ce projet nous a permis de mettre en pratique les concepts théoriques de la méthode des éléments finis et de les appliquer à la résolution numérique d'un problème aux limites. Nous avons acquis une compréhension approfondie de la formulation variationnelle et de la résolution du système linéaire associé, ainsi que des propriétés de la matrice de rigidité.

Nous avons également exploré l'influence du maillage sur la précision de l'approximation, ce qui nous a permis de constater l'importance du choix du maillage pour obtenir des résultats précis. Ce projet a renforcé nos compétences en programmation et en analyse numérique, tout en nous offrant une meilleure compréhension de la résolution des problèmes aux limites à l'aide de la méthode des éléments finis.

Ce travail a consolidé nos connaissances théoriques et pratiques, nous préparant ainsi à aborder des problèmes plus complexes et à utiliser la méthode des éléments finis de manière efficace dans des contextes variés.