TP2:Programmation des éléments finis P1 en 1D

Loubna TALEB

03 Décembre 2023

Contents

1 Introduction

L'objectif de ce TP est de programmer le problème Neumann et de Dirichlet. Ainsi, pour ce faire et suite au dernier TP que j'ai écrit, il nous faut changer les conditions initiales pour les deux problèmes.

2 Problème de Neumann :

2.1 théoriquement:

Posons le problème non homogène suivant :

$$\begin{cases} -u''(x) + u(x) = f(x) & \text{pour } x \in [0,1], \\ u'(0) = u_0 \quad \text{et} \quad u'(1) = u_1. \end{cases}$$

Suite aux travaux précédents, considérons le problème suivant :

$$\begin{cases} \text{Trouver } u \in P^1 \subset H^1([0,1]) \text{ tel que} \\ a(u_h, v_h) = l(v_h) \\ u'(0) = u_0 \quad \text{et} \quad u'(1) = u_1 \end{cases}$$

Tel que $a(u_h, v_h)$ est défini par

$$a(u_h, v_h) = \int_0^1 u_h'(x)v_h'(x)\,dx + \int_0^1 u_h(x)v_h(x)\,dx$$

Le terme $l(v_h)$ est défini par

$$l(v_h) = \int_0^1 f(x)v_h(x)\,dx + [u_h'(x)v_h(x)]_0^1$$

Ainsi,

$$l(v_h) = \int_0^1 f(x)v_h(x)\,dx + u_1v_h(1) - u_0v_h(0)$$

où u_h et v_h sont dans l'espace des éléments finis P^1.
Posons $u_h(x) = \sum_{j=0}^{n} c_j\varphi_j(x)$ et $v_h(x) = \varphi_i(x)$ pour $i = 1, \ldots, n$.
Donc, le problème revient à résoudre $a(u_h, \varphi_i) = l(\varphi_i)$ pour $i = 1, \ldots, n$. Tel que :

$$a(u_h, v_h) = \sum_{j=0}^{n} c_j \left(\int_0^1 \varphi_j'(x)\varphi_i'(x)\,dx + \int_0^1 \varphi_j(x)\varphi_i(x)\,dx \right)$$

et

$$l(\varphi_i) = \int_0^1 f(x)\varphi_i(x)\,dx + u_1\varphi_i(1) - u_0\varphi_i(0)$$

et on a les conditions aux limites :

$$\varphi_i(1) = \begin{cases} 1, & \text{si } i = n, \\ 0, & \text{sinon,} \end{cases}$$

$$\varphi_i(0) = \begin{cases} 1, & \text{si } i = 0, \\ 0, & \text{sinon.} \end{cases}$$

Et en MATLAB, on ajoute les conditions limites comme ceci pour le cas de Neumann :

```
F(n+1) = F(n+1) + u1;
F(1) = F(1) - u0;
```

2.2 solution réel :

Et pour déboguer le code, je vais résoudre le problème par la méthode classique en cherchant une solution particulière. On va prendre le cas simple où $f(x) = 1$.

Pour l'équation homogène, on a $-u''(x) + u(x) = 0$ et on trouve deux racines simples : $r = 1/-1$. Donc, la solution homogène est $u_H = \alpha e^{-x} + \beta e^{x}$.

Pour la solution particulière, c'est clair que $u_p = 1$,
donc $u(x) = \alpha e^{x} + \beta e^{-x} + 1$.

Cherchons donc α et β. On dérive et on remplace par les conditions initiales (on prend $u_0 = 1$ et $u_1 = -1$), on trouve ceci :

$$-\alpha + \beta = 1,$$
$$-\alpha e^{-1} + \beta e = -1.$$

Ainsi, on trouve :

$$\alpha = \frac{1+e}{e^{-1}-e} \quad \text{et} \quad \beta = \frac{e^{-1}+1}{e^{-1}-e}.$$

Et en MATLAB, on programme comme ceci :

```
alpha1=(1+exp(1))/(exp(-1)-exp(1));
beta1=(exp(-1)+1)/(exp(-1)-exp(1));
y1=alpha1*exp(-x)+beta1*exp(x)+1;
```

2.3 résultats pour n=3/5/10/15

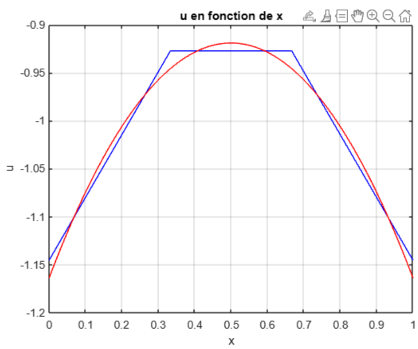


Figure 1: n=3

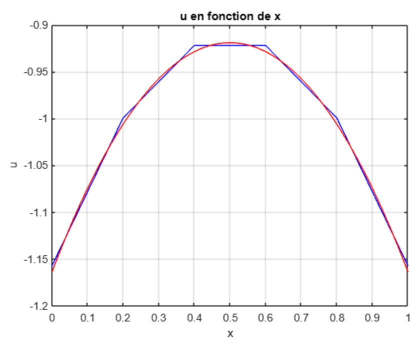


Figure 2: n=5

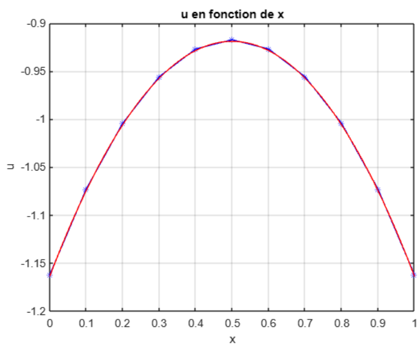


Figure 3: n=10

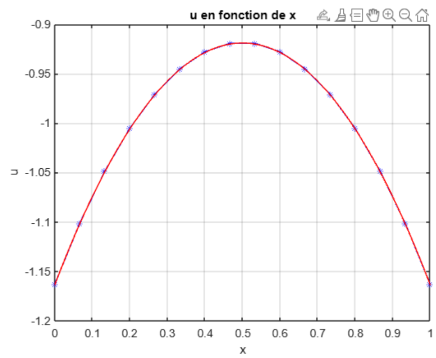


Figure 4: n=15

3 Problème de Dirichlet :

3.1 théoriquement :

Posons le problème non homogène suivant :

$$\begin{cases} -u''(x) + u(x) = f(x) & \text{pour } x \in [0,1], \\ u(0) = u_0 \quad \text{et} \quad u(1) = u_1. \end{cases}$$

donc on a le problème suivant :

$$\begin{cases} \text{Trouver } u \in P^1 \subset H^1([0,1]) \text{ tel que} \\ a(u_h, v_h) = l(v_h) \\ u(0) = u_0 \quad \text{et} \quad u(1) = u_1 \end{cases}$$

Le terme au bord est nul cette fois-ci, comme indiqué dans le polycopié. Afin de prendre en considération les conditions aux limites, nous avons donc le système suivant à résoudre : $Kc = F$, avec pour i,j=1.....n

$$K = \left[K_{ij} = \int_0^1 \varphi_i'(x)\varphi_j'(x)\,dx + \int_0^1 \varphi_j(x)\varphi_i(x)\,dx \right]$$

et

$$c = \left[c_{ij} \right]$$

et

$$F = \left[f_i = \int_0^1 f(x)\varphi_i(x)\,dx \right]$$

Cependant, puisque $u(1) = u_1$ et $u(0) = u_0$, il faut annuler la première colonne et la dernière de la matrice K et aussi celle de la fonction F. Pour ce faire, on va remplacer la première et dernière ligne/colonne par 10^7 aussi pour F. En MATLAB, on programme comme ceci :

```
K(n+1, n+1) = 10^7;
K(1, 1) = 10^7;
F(1) =  u0 * K(1, 1);
F(n+1) = u1 * K(n+1, n+1);
```

3.2 Solution réelle :

Pour déboguer le code, comme pour le problème de Neumann, on va prendre $f(x) = 1$. Ainsi, la solution homogène est $u_H = \alpha e^{-x} + \beta e^{x}$, et la solution particulière est $u_p = 1$. Trouvons α et β tels que $u(x) = \alpha e^{x} + \beta e^{-x} + 1$.

On remplace par les conditions aux limites $u(0) = 1$ et $u(1) = -1$ et on trouve les équations suivantes :

$$\alpha + \beta = 0,$$
$$\alpha e + \beta e^{-1} = -2.$$

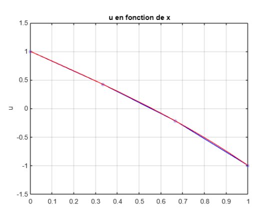
En résolvant ce système, on obtient :

$$\alpha = \frac{-2}{e - e^{-1}}, \quad \beta = \frac{2}{e - e^{-1}}.$$

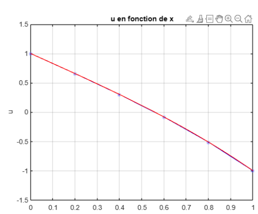
Ainsi, pour programmer cela en MATLAB, on a :

```
alpha2=-2/(exp(1)-exp(-1));
beta2=2/(exp(1)-exp(-1));
y2=alpha2*exp(x)+beta2*exp(-x)+1;
```

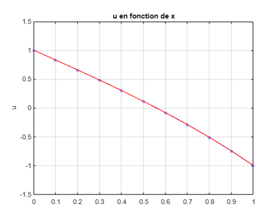
3.3 Résultats pour n=3,n=5,n=10



(a) n=3



(b) n=5



(c) n=10